

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Implementace metody sečen pro nehladkou optimalizaci

On implementation of the secant method for nonsmooth optimization

Zadání diplomové práce

Student: **Bc. Štěpán Bednařík**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 1103T031 Výpočetní matematika

Téma: **Implementace metody sečen pro nehladkou optimalizaci**
On implementation of the secant method for nonsmooth optimization

Jazyk vypracování: čeština

Zásady pro vypracování:

Úloha minimalizace nediferencovatelné (nehladké) funkce se objevuje při řešení některých úloh z mechaniky, např. kontaktní úlohy se třením, či úloh tvarové optimalizace. Tento problém je možné vyřešit buď přímo pomocí algoritmu vhodného pro minimalizaci nediferencovatelných funkcí nebo nepřímo spojitě diferencovatelnou aproximací funkce a následně pomocí algoritmu vhodného pro minimalizaci spojitě diferencovatelných funkcí. Pro minimalizaci nediferencovatelných funkcí bylo vyvinuto v uplynulých desetiletích řada algoritmů. Mezi nejvíce používané patří subgradientní metody a zejména bundle metody. V poslední době se objevují metody, které bundle metodám začínají konkurovat. Mezi ně patří metoda r-sečen. Cílem práce je pochopení této metody a její následná efektivní implementace. V závěru práce se zaměříme na testování algoritmu na vhodných úlohách a jeho porovnání s jinými algoritmy nehladké optimalizace.

Práce bude mít tyto části:

Úloha nehladké optimalizace

Clarkeův kalkul

Algoritmy pro minimalizaci nediferencovatelných funkcí

Metoda r-sečen

Implementace metody r-sečen

Testování algoritmu na vhodných úlohách

Seznam doporučené odborné literatury:

A.Bagirov, A. Ganjehlou: A secant method for nonsmooth optimization. In <http://www.optimization-online.org>, 2007.

J. Zowe: Nondifferentiable optimization: A motivation and a short introduction into the subgradient and the bundle concept. In: NATO SAI Series, 15, Computational Mathematical Programming, Schittkowski, K., (ed.), Springer-Verlag, New York, 1985, pp. 323-356.

F.H. Clarke: Optimization and Nonsmooth Analysis. New York, John Wiley, 1983.

Dále dle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Petr Beremlijski, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017



doc. RNDr. Jiří Bouchala, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

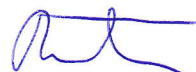
Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 28. dubna 2017



Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2017



Abstrakt

Práce se zabývá metodami pro minimalizaci nehladkých funkcí. Nehladkými funkcemi nazýváme funkce, které nejsou diferencovatelné. Problém minimalizace můžeme řešit nepřímo nahrazením funkce spojitě diferencovatelnou aproximací, nebo pomocí algoritmů, které jsou vhodné pro minimalizaci nediferencovatelných funkcí. Mezi tyto algoritmy patří subgradientní metody, bundle metody a také například metoda r –sečen. V této práci se seznámíme s touto metodou, popíšeme její základní rysy a následně jí otestujeme na vhodných úlohách. Také se seznámíme s nezbytnou teorií z oblasti Clarkeova kalkulu. Navíc porovnáme metodu r –sečen se subgradientními metodami.

Klíčová slova: nehladká optimalizace, nediferencovatelná funkce, subgradient, lokálně lipschitzovsky spojitá funkce, Clarkeův kalkul, metoda r –sečen.

Abstract

This work is devoted to methods for minimization of nonsmooth function. Nonsmooth functions are functions that are not differentiable. Problem of minimization can be solved by replacing original function with continuously differentiable approximation or by using algorithms, that are appropriate for minimization of nonsmooth function. Subdifferential methods and bundle methods are some of them. Another type of algorithm is a r –secant method. The goal of this work is to describe the basics features of this method, to cover the whole algorithm, to implement the code of program and then to test it on suitable problems. This work is also covering the necessary theory of Clarke calculus. On top of that we are also comparing a r –secant method with subdifferential methods.

Key Words: nonsmooth optimization, nondifferentiable function, subdifferential, Lipschitz function, Clarke calculus, r –secant method.

Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam tabulek	11
Seznam výpisů zdrojového kódu	12
1 Úvod	13
2 Metody nehladké optimalizace	15
3 Clarkeův kalkul	16
3.1 Lipschitzovská spojitost	16
3.2 Konvexní obal	16
3.3 Rademacherův teorém	18
3.4 Zobecněná směrová derivace a zobecněný gradient	18
3.5 Lokální extrém	20
3.6 Polohladkost a regulárnost funkce	21
4 Subgradientní metody	22
4.1 První varianta	22
4.2 Druhá varianta	24
4.3 Třetí varianta	27
5 Metoda r–sečen	30
5.1 r –sečna	30
5.2 Nutné podmínky minima	36
5.3 Směr poklesu	39
6 Popis algoritmu r–sečen	41
6.1 Algoritmus 1	41
6.2 Algoritmus 2	43
6.3 Algoritmus 3	44

7	Numerické experimenty	46
7.1	S funkcí jedné proměnné	46
7.2	S funkcemi dvou proměnných	50
7.3	S pružnou membránou	63
8	Závěr	68
	Literatura	69

Seznam použitých zkratk a symbolů

$ \cdot $	– absolutní hodnota
$\ \cdot\ $	– Eukleidovská norma
$\langle\cdot,\cdot\rangle$	– skalární součin
\bigcup	– sjednocení množin
0_n	– nulový vektor dimenze n
B_1	– jednotková sféra v R^n se středem v 0_n
B_δ	– sféra v R^n se středem v 0_n a poloměrem δ
$\nabla f(x)$	– gradient funkce
$f^0(x;v)$	– Clarkeova zobecněná směrová derivace
$\partial f(x)$	– Clarkeův zobecněný gradient
argmax	– argument maxima
$\operatorname{conv} X$	– konvexní obal
\limsup	– limes superior
ε	– tolerance pro ukončující podmínku
f_{opt}	– optimální funkční hodnota
x_0, u_0	– počáteční bod, počáteční vektor

Seznam obrázků

1	Konvexní obal množiny M ($\text{conv } M$)	17
2	Graf funkce $f(x) = x $	19
3	Zobecněný gradient $\partial f(x)$ pro funkci $f(x) = x $	20
4	Konvergenční historie subgradientní metody (varianta 1) pro funkci $f(x) = x $ s numericky vypočteným gradientem	23
5	Konvergenční historie subgradientní metody (varianta 1) pro funkci $f(x) = x $ s analyticky vypočteným gradientem	24
6	Konvergenční historie subgradientní metody (varianta 2) pro funkci $f(x) = x $ s numericky vypočteným gradientem	26
7	Konvergenční historie subgradientní metody (varianta 2) pro funkci $f(x) = x $ s analyticky vypočteným gradientem	26
8	Konvergenční historie subgradientní metody (varianta 3) pro funkci $f(x) = x $ s numericky vypočteným gradientem	28
9	Konvergenční historie subgradientní metody (varianta 3) pro funkci $f(x) = x $ s analyticky vypočteným gradientem	29
10	Zobecněný gradient $\partial f(0, 0)$ pro funkci $f(x_1, x_2) = x_1 + x_2 $	36
11	Množina $S_1^c f(x)$ pro funkci $f(x) = x $	38
12	Množina $S_1^c f(x) + B_{0.5}$ pro funkci $f(x) = x $	39
13	Konvergenční historie metody r –sečen pro funkci $f(x) = x $ s parametry $x_0 = 1, r = 5$	48
14	Konvergenční historie metody r –sečen pro funkci $f(x) = x $ s parametry $x_0 = 1, r = 10$	49
15	Konvergenční historie metody r –sečen pro funkci $f(x) = x $ s parametry $x_0 = -2, r = 5$	49
16	Konvergenční historie metody r –sečen pro funkci $f(x) = x $ s parametry $x_0 = -2, r = 10$	50
17	Graf funkce $f(x_1, x_2) = x_1 + x_2 $	51
18	Konvergence posloupnosti x^k metody r –sečen pro funkci $f(x_1, x_2) = x_1 + x_2 $	52
19	Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = x_1 + x_2 $ s parametry $x_0 = (1, -2), r = 5, \varepsilon = 10^{-3}$	52
20	Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = x_1 + x_2 $ s parametry $x_0 = (1, -2), r = 5, \varepsilon = 10^{-4}$	53
21	Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = x_1 + x_2 $ s parametry $x_0 = (-2, 3), r = 5, \varepsilon = 10^{-3}$	53
22	Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = x_1 + x_2 $ s parametry $x_0 = (-2, 3), r = 5, \varepsilon = 10^{-4}$	54
23	Graf funkce $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	55

24	Konvergence posloupnosti x^k metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	56
25	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (1, -2)$, $r = 5$, $\varepsilon = 10^{-3}$	56
26	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (1, -2)$, $r = 10$, $\varepsilon = 10^{-3}$	57
27	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (-2, 3)$, $r = 5$, $\varepsilon = 10^{-3}$	57
28	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (-2, 3)$, $r = 10$, $\varepsilon = 10^{-3}$	58
29	Graf funkce $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$.	59
30	Konvergence posloupnosti x^k metody r -sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$	60
31	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.5$	61
32	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.4$	61
33	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.3$	62
34	Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.2$	62
35	Popis pružné membrány	63
36	Prohnutí membrány pro $g = 1$	66
37	Prohnutí membrány pro $g = 0.3$	66
38	Prohnutí membrány pro $g = 0.1$	67

Seznam tabulek

1	Výsledky první varianty subgradientní metody	23
2	Výsledky druhé varianty subgradientní metody	25
3	Výsledky třetí varianty subgradientní metody	28
4	Výsledky metody r –sečen pro funkci $f(x) = x $	47
5	Porovnání subgradientní metody a metody r –sečen pro numerický výpočet subgradientu	47
6	Porovnání subgradientní metody a metody r –sečen pro analytický výpočet subgradientu	47
7	Výsledky metody r –sečen pro funkci $f(x_1, x_2) = x_1 + x_2 $	51
8	Výsledky metody r –sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$. . .	55
9	Výsledky metody r –sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$	60
10	Výsledky metody r –sečen pro řešení úlohy prohnutí membrány	65

Seznam výpisů zdrojového kódu

1	První varianta subgradientní metody	22
2	Druhá varianta subgradientní metody	25
3	Třetí varianta subgradientní metody	27
4	Algoritmus 1 (metoda r –sečen)	42
5	Algoritmus 2 (metoda r –sečen)	44
6	Algoritmus 3 (metoda r –sečen)	45

1 Úvod

Úkolem diplomové práce bylo seznámení se s metodou r –sečen a její následná implementace. Dalším úkolem bylo testování algoritmu na vhodných úlohách. Tato metoda patří mezi metody nehladké optimalizace, které se zabývají minimalizací nediferencovatelných funkcí. O těchto funkcích předpokládáme, že jsou pouze spojitě. Nemusí být zaručena jejich diferencovatelnost.

Tento text se zabývá popisem, následným sestavením a praktickým použitím algoritmu, ale také popisem teorie, která se v algoritmu využívá. V textu se také objevují další metody vhodné pro řešení nehladkých optimalizací.

V druhé kapitole je definován námi řešený matematický problém s popisem jeho specifických vlastností, mezi které patří např. předpoklad, že nemusí být zaručena diferencovatelnost funkce. Jsou zde také navrženy možné metody řešení daného problému (metody nultého řádu, nepřímé metody).

Třetí kapitola obsahuje teorii, která je potřeba k řešení iteračních metod určených pro minimalizaci nehladkých funkcí. Zavádíme zde pojmy jako lipschitzovská spojitost, konvexní obal, zobecněná směrová derivace, zobecněný gradient, polohladkost a regulárnost funkce. Také zde uvádíme Rademacherův teorém a kritéria lokálního extrému. Pojmy zavedeny v této kapitole jsou poté ilustrovány na příkladech.

Ve čtvrté kapitole se zabýváme konkrétní metodou pro řešení optimalizace nehladkých funkcí - subgradientní metodou. Nejdříve popíšeme základní vlastnosti této metody a poté se zaměříme na tři různé varianty subgradientní metody. Ukážeme si, jak vypadají algoritmy, které se používají k výpočtům a jaké specifika mají jednotlivé varianty. Následují numerické testy těchto metod s výsledky v tabulkách a také grafy, které popisují tzv. konvergenční historie.

Následuje pátá kapitola, která se zabývá metodou r –sečen z pohledu teorie. Zde se dozvíme, co znamená pojem r –sečna a k čemu se dá použít. Jsou zde definovány i množiny, které se využívají při optimalizaci, určují se zde nutné podmínky pro minimum funkce a také se zde zavádějí různé typy stacionárních bodů. Tyto pojmy jsou poté opět ilustrovány na příkladech. Dozvíme se zde rovněž, jak se počítá optimální směr poklesu cenové funkce.

Šestá kapitola obsahuje popis algoritmu r –sečen. Algoritmus obsahuje tři části, které jsou každá popsána v samostatné podkapitole. V podkapitolách jsou obsaženy i pseudokódy, které jednotlivé části algoritmu popisují. Každá z podkapitol též obsahuje předpoklady o konvergenci jednotlivých částí algoritmu. Je zde také uvedena ukončující podmínka algoritmu.

V poslední (sedmé) kapitole testujeme algoritmus na konkrétních příkladech. Nejprve porovnááme výsledky algoritmu r –sečen a tří variant subgradientní metody na funkci jedné proměnné. Poté testujeme algoritmus na funkcích dvou proměnných. Zde je vždy nejdříve uveden graf funkce, poté tabulka s výsledky a grafy konvergenčních historií, společně se závěrem, který z výsledků vyplývá. Ukazujeme si tedy také grafy, které popisují konvergenci posloupnosti iteračních bodů generovaných metodou r –sečen k minimu funkce. Poslední část numerických experimentů je věnována modelové úloze prohnutí membrány. Zde si ukazujeme jak se dá použít algoritmus r –sečen k řešení úlohy, která popisuje prohnutí membrány, na kterou na hranici navíc působí odpor prostředí.

2 Metody nehladké optimalizace

Metody nehladké optimalizace se zabývají minimalizací funkcí, o kterých předpokládáme, že jsou pouze spojité. Není tedy zaručena diferencovatelnost funkce.

Úlohu můžeme zapsat ve tvaru

$$\min_{x \in \Omega} f(x), \tag{1}$$

kde je funkce $f : R^n \rightarrow R$ a pro množinu Ω platí $\Omega \subset R^n$. Funkce f je navíc pouze lokálně lipschitzovsky spojitá (tento pojem si zavedeme později) a není spojitě diferencovatelná.

Tyto úlohy můžeme řešit různými metodami. Jeden z typů jsou tzv. metody nultého řádu, které pro svůj běh nepotřebují znalost gradientu ani Hessiánu. Mezi tyto metody se řadí tzv. genetické algoritmy. Další z možností řešení jsou nepřímé metody, které místo funkce, která není spojitě diferencovatelná, minimalizují její diferencovatelnou aproximaci, anebo iterační metody určené přímo pro minimalizaci funkcí, které nejsou spojitě diferencovatelné. Zde můžeme zařadit hlavně subgradientní metody, které využívají znalost jednoho libovolného subgradientu v aktuální iteraci (pojem subgradient vysvětlíme později), a svazkové (bundle) metody, které využívají znalosti svazku subgradientů v aktuální iteraci i v iteracích předcházejících.

Nevýhodou metod nultého řádu je jejich nevhodnost pro řešení komplikovanějších optimalizačních úloh, kde se pracuje se složitější funkcí. Pokud je výpočet funkčních hodnot této funkce „drahý“, pak je použití těchto metod extrémně časově náročné, protože pro metody nultého řádu je obvyklý výpočet velkého počtu funkčních hodnot.

Nepřímé metody sice mohou být rychlé, ale jejich slabinou je minimalizace funkce, která pouze aproximuje původní spojitě nediferencovatelnou funkci.

Více o metodách nehladké optimalizace můžete nalézt v [4] a [7].

3 Clarkeův kalkul

Pro řešení našich úloh budeme používat iterační metody, které jsou přímo určeny pro minimalizaci nehladkých funkcí. Abychom mohli tyto metody zkonstruovat, musíme nejdříve zavést tzv. Clarkeův kalkul. Pomocí něj získáme aproximaci funkce i v bodě, ve kterém není diferencovatelná. Tuto aproximaci nazveme subgradient. Clarkeův kalkul je pro své vlastnosti vhodný k algoritmicizaci hledání minima funkcí, jež nejsou spojitě diferencovatelné.

Veškeré důkazy následujících tvrzení a více informací může čtenář nalézt v [3] a [4].

3.1 Lipschitzovská spojitost

Definice 1 *Nechť množina $M \subset R^n$, funkce $f : R^n \rightarrow R$ a K je nezáporné reálné číslo. Nechť je pro funkci f splněna podmínka:*

$$|f(x') - f(x'')| \leq K \|x' - x''\| \quad \forall x', x'' \in M, \quad (2)$$

*pak řekneme, že funkce f je **lipschitzovsky spojitá (s modulem K) na množině M .***

Definice 2 *Nechť je pro funkci f splněna podmínka:*

$$\exists \epsilon > 0 : |f(x') - f(x'')| \leq K \|x' - x''\| \quad \forall x', x'' \in x + \epsilon B_1, \quad (3)$$

*kde B_1 je jednotková sféra. Pak řekneme, že funkce f je **lipschitzovsky spojitá v bodě $x \in R^n$ (s modulem K).***

Definice 3 *Nechť je funkce f lipschitzovsky spojitá v okolí každého $x \in R^n$. Pak řekneme, že funkce f je **lokálně lipschitzovsky spojitá na R^n .***

Například funkce $f(x) = x^2$ je lipschitzovsky spojitá na intervalu $\langle 0, 1 \rangle$, ale není lipschitzovsky spojitá na R . Je tedy pouze lokálně lipschitzovsky spojitá na R .

3.2 Konvexní obal

Definice 4 *Nechť $X = \{x_1, \dots, x_m\} \subseteq R^n$. Konvexní obal množiny X , který označujeme jako $\text{conv } X$, je množina definovaná předpisem*

$$\text{conv } X = \left\{ \sum_{i=1}^m \lambda_i x_i \mid x_1, \dots, x_m \in X, \lambda_1, \dots, \lambda_m \geq 0, \sum_{i=1}^m \lambda_i = 1 \right\}, \quad (4)$$

Tzn. že $\text{conv } X$ je roven množině všech konvexních (nezáporných, afinních) kombinací proků z X .

3.2.1 Příklad

Pro ilustraci pojmu konvexní obal uvedeme jednoduchý příklad. Mějme množinu $M = \{1, 2\} \in \mathbb{R}$. Z (4) víme, že

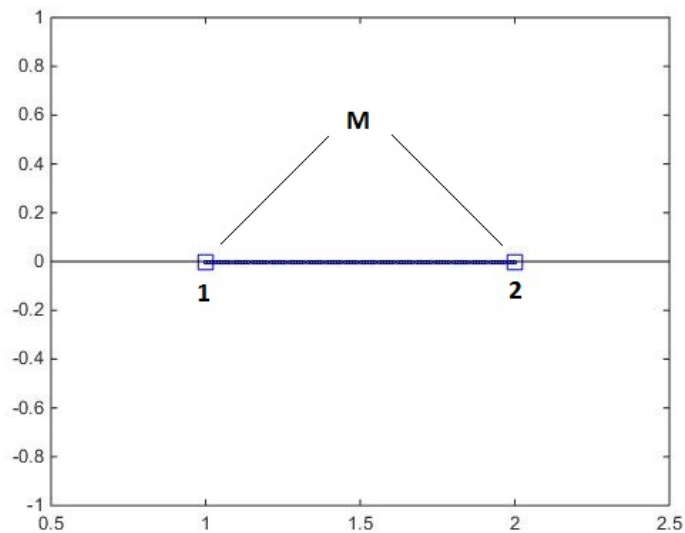
$$\text{conv}\{1, 2\} = \lambda_1 \cdot 1 + \lambda_2 \cdot 2, \quad (5)$$

a platí, že $\lambda_{1,2} \geq 0$, $\lambda_1 + \lambda_2 = 1$.

Ze součtu $\lambda_1 + \lambda_2 = 1$ vyjádříme $\lambda_2 = 1 - \lambda_1$ a dosadíme do (5). Výsledný konvexní obal tedy vypadá následovně

$$\text{conv}\{1, 2\} = \lambda_1 + (1 - \lambda_1) \cdot 2 = 2 - \lambda_1, \lambda_1 \in \langle 0, 1 \rangle$$

Konvexní obal množiny $M = \{1, 2\} \in \mathbb{R}$ je tedy interval $\langle 1, 2 \rangle$. Uvedeme ještě obrázek pro geometrickou ilustraci pojmu.



Obrázek 1: Konvexní obal množiny M ($\text{conv } M$)

Obrázek ilustruje, že jsme dostali celý interval $\langle 1, 2 \rangle$. Pro $n = 1$ je totiž konvexní obal množiny, která obsahuje dva různé body, interval mezi těmito body.

3.3 Rademacherův teorém

Věta 1 *Nechť funkce $f : R^n \rightarrow R$ je lokálně lipschitzovsky spojitá na R^n a nechť $D = \{x \in R^n \mid f \text{ je diferencovatelná v } x\}$. Potom Lebesguova míra množiny bodů, ve kterých funkce f není diferencovatelná, tj. množiny $(R^n \setminus D)$, je nula.*

3.4 Zobecněná směrová derivace a zobecněný gradient

Definice 5 *Nechť funkce $f : R^n \rightarrow R$ je lipschitzovsky spojitá v okolí bodu x a nechť v je vektor v R^n . **Clarkeova zobecněná směrová derivace** funkce f v bodě x ve směru v , kterou značíme $f^0(x; v)$, je definovaná předpisem:*

$$f^0(x; v) = \limsup_{y \rightarrow x, t \downarrow 0} \frac{f(y + tv) - f(y)}{t}, \quad (6)$$

kde y je vektor v R^n a t je kladné reálné číslo.

Definice 6 *Nechť funkce $f : R^n \rightarrow R$ je lipschitzovsky spojitá v okolí bodu x . **Clarkeův zobecněný gradient** funkce f v bodě x , který značíme $\partial f(x)$, je množina*

$$\partial f(x) = \{\zeta \in R^n \mid f^0(x; v) \geq \langle \zeta, v \rangle \forall v \in R^n\}. \quad (7)$$

Prvky $\partial f(x)$ nazýváme **Clarkeovými subgradienty** funkce f v bodě x .

Díky Rademacherovu teorému (u lokálně lipschitzovských spojitých funkcí máme zaručenu diferencovatelnost téměř všude ve smyslu Lebesguovy míry) máme také jinou možnost, jak zkonstruovat Clarkeův zobecněný gradient. Tuto alternativní možnost vidíme níže.

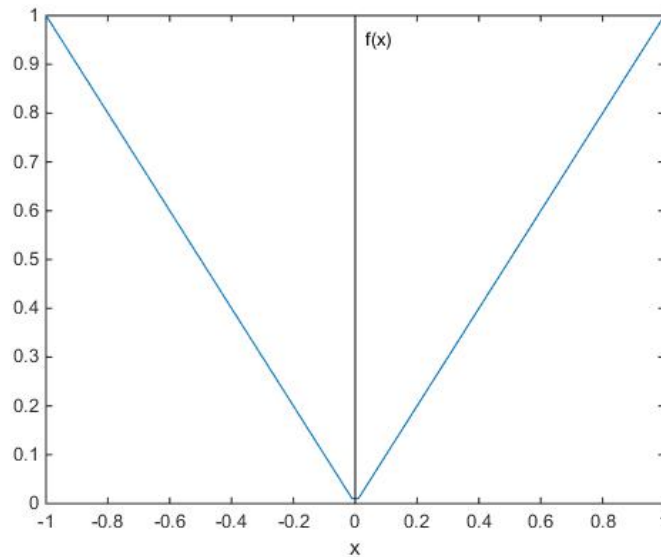
Věta 2 *Nechť funkce $f : R^n \rightarrow R$ je lipschitzovsky spojitá v okolí bodu x . Potom*

$$\partial f(x) = \text{conv}\left\{\lim_{i \rightarrow \infty} \nabla f(x_i) \mid x_i \rightarrow x, x_i \notin \Omega_f\right\}, \quad (8)$$

kde $\Omega_f = \{x \in R^n \mid f \text{ není diferencovatelná v } x\}$ a conv značí konvexní obal.

3.4.1 Příklad

Jako příklad k ilustraci Clarkeova zobecněného gradientu použijeme funkci $f(x) = |x|$. Tato funkce je lokálně lipschitzovsky spojitá na \mathbb{R} a proto diferencovatelná skoro všude.



Obrázek 2: Graf funkce $f(x) = |x|$

Na obrázku vidíme, že jediným problémovým bodem bude $x = 0$, kde dochází k nespojitosti první derivace.

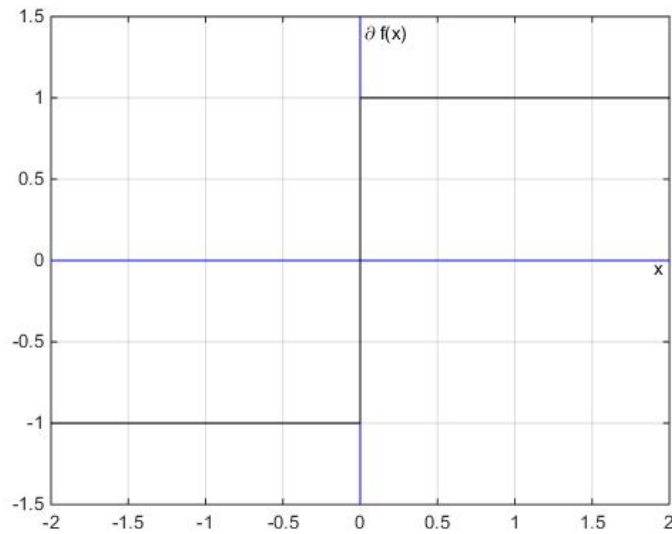
Víme, že:

- $f'(x) = -1$ pro $x < 0$
- $f'(x) = 1$ pro $x > 0$

Nyní musíme spočítat zobecněný gradient funkce $f(x) = |x|$ pro $x = 0$.

$$\begin{aligned}\partial f(0) &= \text{conv}\left\{\left\{\lim_{i \rightarrow \infty} f'(x_i) \mid x_i \rightarrow 0_+\right\} \cup \left\{\lim_{i \rightarrow \infty} f'(x_i) \mid x_i \rightarrow 0_-\right\}\right\} \\ &= \text{conv}\left\{\left\{\lim_{i \rightarrow \infty} 1\right\} \cup \left\{\lim_{i \rightarrow \infty} -1\right\}\right\} = \text{conv}\{1, -1\} = \langle -1, 1 \rangle\end{aligned}$$

Graf zobecněného gradientu $\partial f(x)$ funkce $f(x) = |x|$ je na následujícím obrázku



Obrázek 3: Zobecněný gradient $\partial f(x)$ pro funkci $f(x) = |x|$

3.5 Lokální extrém

Pro hledání lokálních extrémů funkcí, které nejsou spojitě diferencovatelné, potřebujeme nástroj, který nám umožní rozpoznat, zda funkce f má v bodě x lokální extrém.

Věta 3 *Jestliže funkce f , která je lokálně lipschitzovsky spojitá na R^n , nabývá v bodě x svého lokálního maxima nebo minima, potom platí*

$$0_n \in \partial f(x) \quad (9)$$

Body splňující výše uvedenou podmínku nazýváme **Clarkeovými stacionárními body** optimalizační úlohy, zavedené na začátku kapitoly.

Z příkladu 3.4.1 vidíme, že $0 \in \partial f(0)$. Funkce $f(x) = |x|$ nabývá svého lokálního extrému v bodě $x = 0$. Z grafu funkce dokonce víme, že tento bod je i globálním minimem funkce.

3.6 Polohladkost a regulárnost funkce

Definice 7 O funkci $f : R^n \rightarrow R$ řekneme, že je **polohladká** v bodě x , jestliže funkce f je lipschitzovsky spojitá v okolí bodu x a dále platí, že pro všechny $v \in R^n$ existuje limita

$$\lim_{V \in \partial f(x+tv'), v' \rightarrow v, t \downarrow 0} \{Vv'\} \quad (10)$$

Definice 8 O funkci $f : R^n \rightarrow R$ řekneme, že je **slabě polohladká** v bodě x , jestliže funkce f je lipschitzovsky spojitá v okolí bodu x a dále platí, že pro všechny $v \in R^n$ existuje limita

$$\lim_{V \in \partial f(x+tv), t \downarrow 0} \{Vv\} \quad (11)$$

Je zřejmé, že polohladkost funkce v bodě implikuje slabou polohladkost.

Věta 4 Nechť funkce $f : R^n \rightarrow R$ je slabě polohladká v bodě x . Potom směrová derivace

$$f'(x; v) = \lim_{t \downarrow 0} \frac{f(x + tv) - f(x)}{t} \quad (12)$$

existuje pro všechny $v \in R^n$ a platí

$$f'(x; v) = \lim_{V \in \partial f(x+tv), t \downarrow 0} \{Vv\}. \quad (13)$$

Definice 9 O funkci f řekneme, že je **regulární** v bodě x , pokud splňuje následující podmínky

1. Existují směrové derivace $f'(x; v)$ pro všechny vektory $v \in R^n$.
2. Platí rovnost $f'(x; v) = f^0(x; v)$ pro všechny vektory $v \in R^n$.

Všimněme si, že podmínka 2 z předcházející definice je nutná. Lze najít funkce, pro které existuje v bodě x směrová derivace $f'(x; v)$ a přitom podmínka 2 není splněna, např. pro funkci $f(x) = -|x|$ platí $f^0(x; v) \neq f'(x; v)$.

4 Subgradientní metody

Tyto metody jsou jednou z možností pro řešení úloh nehladké optimalizace. Subgradientní metoda vyžaduje v každém kroku vypočítat funkční hodnotu funkce $f(x)$ a jeden libovolný Clarkeův subgradient. Na základě těchto znalostí pak metoda vygeneruje jednotkový směr kroku d pro stávající iteraci. Hlavní nevýhodou je neexistence „rozumné“ implementovatelné ukončující podmínky a neexistence dobré strategie pro určování délky kroku t_k , protože směr d nemusí být směr poklesu. Volba délky kroku je tedy provedena předem.

Dá se ukázat, že délka kroku $t_k, k \in N$ musí splňovat následující omezení:

- $t_k > 0$
- $\lim t_k = 0$
- $\sum_{k=0}^{\infty} t_k = \infty$

Protože se subgradientní metoda vyznačuje velmi nízkou rychlostí konvergence, není vhodná pro řešení složitějších úloh nehladké optimalizace. Existuje několik možností, jak určit délku kroku t_k . Ukážeme si algoritmy a výsledky pro některé z nich. Více informací na toto téma lze nalézt v [4] a [7].

4.1 První varianta

Podle podmínek volíme $t_k = \frac{1}{k}$. Dále si vybereme libovolný subgradient $g_k \in \partial f(x)$.

Algoritmus metody poté vypadá takto:

```
Input  $x_0, \varepsilon$ 
while  $\|x_{k+1} - x_k\| > \varepsilon$ 
     $d_k = -\frac{g_k}{\|g_k\|}$ 
     $t_k = \frac{1}{k}$ 
     $x_{k+1} = x_k + t_k \cdot d_k$ 
end
Output  $x_k$  - aproximace minima
```

Výpis 1: První varianta subgradientní metody

Subgradient $g_k \in \partial f(x)$ můžeme použitím numerické derivace odhadnout s pomocí vzorce:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, h > 0 \quad (14)$$

anebo přesně vypočítat pomocí analyticky určené derivace.

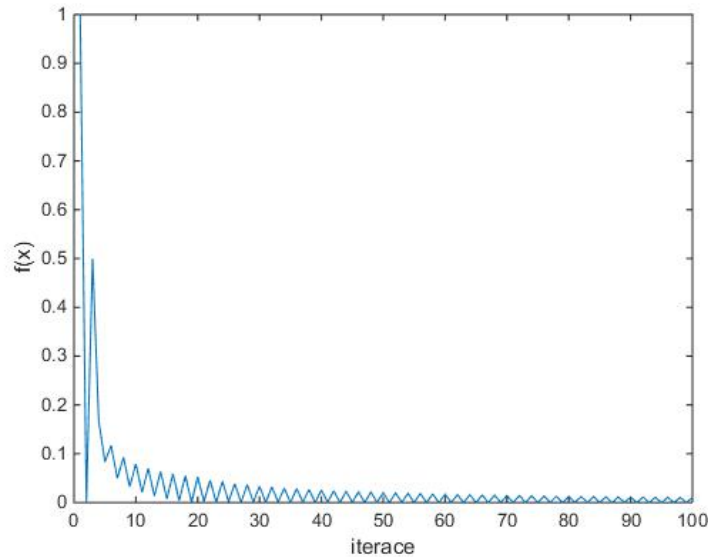
Uvedený algoritmus jsme testovali opět na funkci $f(x) = |x|$. V tabulce vidíme rozdílné hodnoty nalezeného řešení pro volbu různých počátečních bodů a přesností v závislosti na volbě metody výpočtu subgradientu (numericky, analyticky).

Tabulka 1: Výsledky první varianty subgradientní metody

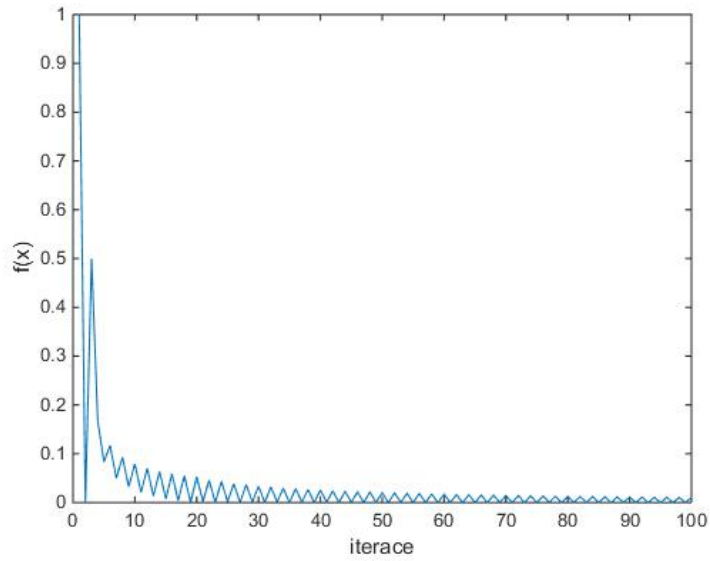
x_0	ε	Aproximace minima x^* (subgradient vypočten numericky)	Aproximace minima x^* (subgradient vypočten analyticky)
1	0,1	0.0210	-0.0210
1	0,01	$9.1936 \cdot 10^{-5}$	$-9.6865 \cdot 10^{-5}$
1	0,001	$-3.9990 \cdot 10^{-6}$	$1.0010 \cdot 10^{-6}$
1	0,0001	$-4.9900 \cdot 10^{-6}$	$1.0001 \cdot 10^{-6}$
-2	0,1	0.0210	0.0210
-2	0,01	$9.1936 \cdot 10^{-5}$	$9.6865 \cdot 10^{-5}$
-2	0,001	$-3.9990 \cdot 10^{-6}$	$-1.0010 \cdot 10^{-6}$
-2	0,0001	$-4.9900 \cdot 10^{-6}$	$1.0001 \cdot 10^{-6}$

Z výsledků vidíme, že čím menší ε zvolíme, tím více se výsledná aproximace blíží minimu funkce, které je v bodě $x = 0$.

Jak jsme si řekli na začátku kapitoly, hlavní nevýhodou je, že směr d nemusí být vždy směr poklesu. Tuto vlastnost si nyní ukážeme na grafech tzv. konvergenčních historií, které ukazují závislost funkční hodnoty na současném bodě iterace. Budeme opět uvažovat funkci $f(x) = |x|$ a $x_0 = 1, \varepsilon = 0,01$.



Obrázek 4: Konvergenční historie subgradientní metody (varianta 1) pro funkci $f(x) = |x|$ s numericky vypočteným gradientem



Obrázek 5: Konvergenční historie subgradientní metody (varianta 1) pro funkci $f(x) = |x|$ s analyticky vypočteným gradientem

Na grafech konvergenční historie si můžeme potvrdit, že krok d nemusí být vždy směr poklesu. Vidíme také, že v případě volby kroku $t_k = \frac{1}{k}$ máme vždy i dopředu přesně daný počet iterací. Počet iterací je dán jako obrácená hodnota pro přesnost ε . Zde jsme dostali pro volbu $\varepsilon = 0,01$ přesně 100 iterací.

4.2 Druhá varianta

Předpokládejme, že známe funkční hodnotu minima $f(x^*)$ pro libovolné optimální x^* . Pokud navíc pro $L > 0$ platí, že $f(x) - f(x^*) \geq L\|x - x^*\|$ pro všechna x , můžeme ideální krok t_k vypočítat takto:

$$t_k = \lambda \frac{f(x_k) - f(x^*)}{\|g_k\|}, \quad (15)$$

kde $0 < \lambda < 2$. Díky takto zvolené posloupnosti kroků algoritmus konverguje do x^* .

Algoritmus poté může vypadat následovně:

```

Input  $x_0, \varepsilon, f(x^*), \lambda$ 
while  $\|x_{k+1} - x_k\| > \varepsilon$ 
     $d_k = -\frac{g_k}{\|g_k\|}$ 
     $t_k = \lambda \frac{f(x_k) - f(x^*)}{\|g_k\|}$ 
     $x_{k+1} = x_k + t_k \cdot d_k$ 
end
Output  $x_k$  - aproximace minima

```

Výpis 2: Druhá varianta subgradientní metody

Zde g_k opět představuje libovolný subgradient dané funkce.

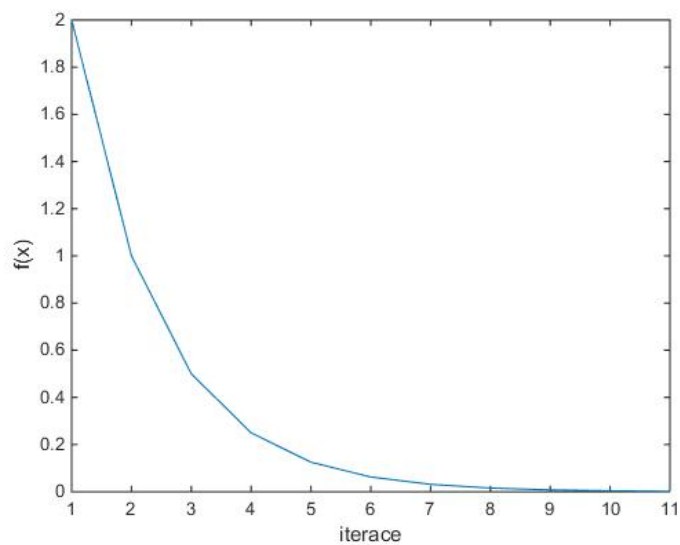
Výsledky a počet iterací závisící na volbě různých počátečních parametrů pro funkci $f(x) = |x|$ můžeme porovnat v tabulce:

Tabulka 2: Výsledky druhé varianty subgradientní metody

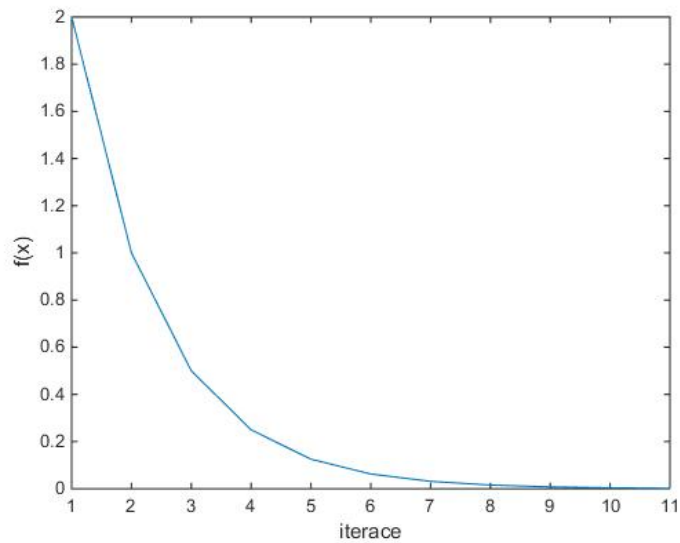
x_0	ε	λ	Aproximace minima x^* (subgradient vypočten numericky)		Aproximace minima x^* (subgradient vypočten analyticky)	
			x^*	Počet iterací	x^*	Počet iterací
1	0,1	0,5	0.0625	5	0.0625	5
1	0,01	1	0	3	0	3
1	0,001	1,5	$2.4414 \cdot 10^{-4}$	13	$2.4414 \cdot 10^{-4}$	13
-2	0,1	0,5	-0.0625	6	-0.0625	6
-2	0,01	1	$-2.6205 \cdot 10^{-11}$	3	0	3
-2	0,001	1,5	$2.4414 \cdot 10^{-4}$	14	$2.4414 \cdot 10^{-4}$	14

Vidíme, že přesnost opět záleží na volbě počátečních parametrů (přesnosti a tentokrát i λ). Výsledné aproximace jsou velmi podobné bez závislosti na volbě možnosti výpočtu subgradientu. Z počtu iterací lze také vidět, že má tato varianta subgradientní metody daleko větší rychlost konvergence oproti první variantě, kde byl počet iterací přesně dán jako obrácená hodnota pro přesnost ε . Ve všech šesti případech jsme volili $f(x^*) = 0$.

Ukážeme si opět grafy konvergenčních historií, tentokrát pro volbu počátečních parametrů $x_0 = -2, \varepsilon = 0,001$ a $\lambda = 0,5$.



Obrázek 6: Konvergenční historie subgradientní metody (varianta 2) pro funkci $f(x) = |x|$ s numericky vypočteným gradientem



Obrázek 7: Konvergenční historie subgradientní metody (varianta 2) pro funkci $f(x) = |x|$ s analyticky vypočteným gradientem

Pro zadanou volbu parametrů algoritmus skončil po 12 iteracích. Z grafů vidíme velkou rychlost konvergence této varianty a také, že směr d je pro tyto parametry vždy směrem poklesu.

4.3 Třetí varianta

Uvedeme ještě třetí možnost pro hledání minima nehladké funkce pomocí subgradientní metody. V této alternativě se používá aktualizace tzv. dilatační matice H , která je v každém kroku upravena pomocí vzorce:

$$H_{k+1} = a_k(H_k - B_k d_k d_k^T),$$

kde $d_k = H_k g_k / \sqrt{g_k^T H_k g_k}$ a a_k, B_k, t_k jsou kladná reálná čísla, vypočtena jako:

$$a_k = \frac{n^2}{n^2 - 1}, B_k = \frac{2}{n + 1}, t_k = \frac{1}{n + 1}$$

Připomeňme, že n je dimenze funkce f .

Nový iterační bod poté hledáme takto:

$$x_{k+1} = x_k - t_k H_k g_k / \sqrt{g_k^T H_k g_k} = x_k - t_k d_k$$

V první iteraci volíme matici $H_0 = aI$. Není těžké si uvědomit, že ve speciálním případě volby $a = 1, a_k = 1, B_k = 0$ dostáváme stejný algoritmus jako pro první variantu subgradientní metody.

Algoritmus třetí varianty vypadá takto:

```
Input  $x_0, \varepsilon$ 
if  $n == 1$ 
     $a = 1$ 
else
     $a = \frac{n^2}{n^2 - 1}$ 
end;
 $B = \frac{2}{n+1}$ 
 $t_k = \frac{1}{n+1}$ 
while  $\|x_{k+1} - x_k\| > \varepsilon$ 
     $d_k = H_k g_k / \sqrt{g_k^T H_k g_k}$ 
     $x_{k+1} = x_k - t_k \cdot d_k$ 
     $H_{k+1} = a_k(H_k - B_k d_k d_k^T)$ 
end
Output  $x_k$  - aproximace minima
```

Výpis 3: Třetí varianta subgradientní metody

g_k je opět libovolný subgradient pro danou funkci.

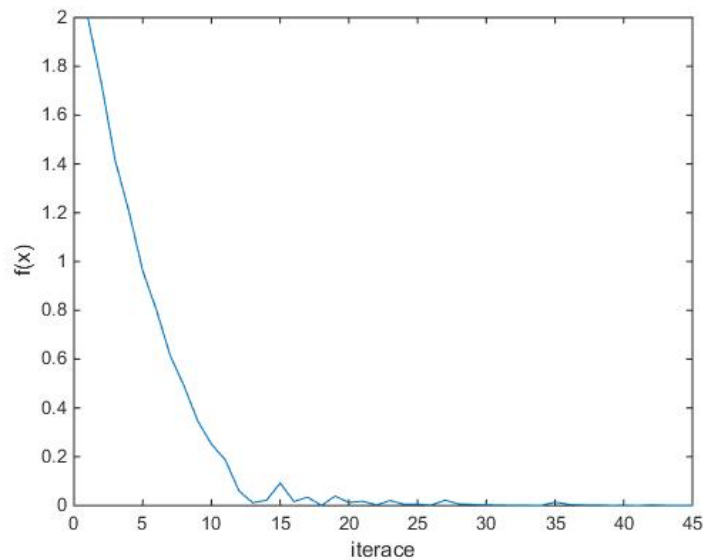
Budeme opět zkoumat rozdílnost výsledků a počet iterací při různé volbě počátečních parametrů. Volíme znovu funkci $f(x) = |x|$. Výsledky vidíme v následující tabulce.

Tabulka 3: Výsledky třetí varianty subgradientní metody

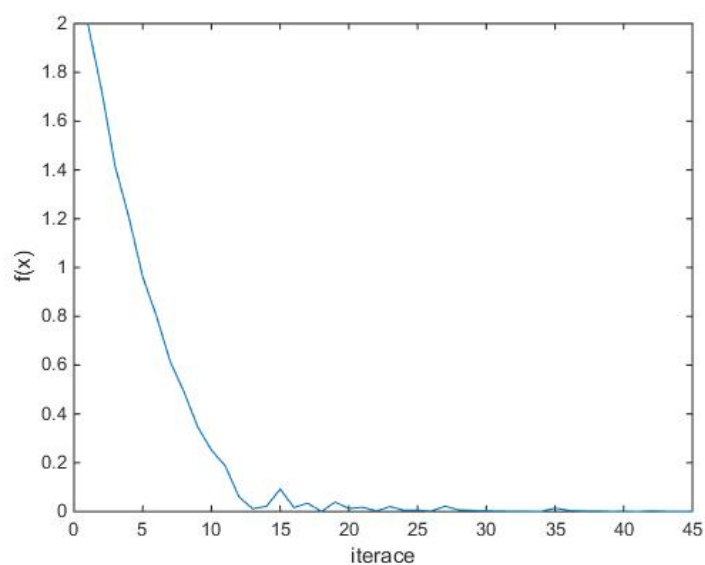
x_0	ε	Aproximace minima x^* (subgradient vypočten numericky)		Aproximace minima x^* (subgradient vypočten analyticky)	
		x^*	Počet iterací	x^*	Počet iterací
1	0,1	-0.0239	11	-0.0239	11
1	0,01	-0.0011	29	-0.0011	29
1	0,001	$0.0169 \cdot 10^{-3}$	40	$0.0169 \cdot 10^{-3}$	40
-2	0,1	-0.1878	11	-0.1878	11
-2	0,01	0.0017	26	0.0017	26
-2	0,001	$0.0463 \cdot 10^{-3}$	46	$0.0463 \cdot 10^{-3}$	46

Zde vidíme, že na výsledek nemá vliv volba pro výpočet subgradientu. Počet iterací pro tuto variantu je menší, než u první varianty, což nám ukazuje větší rychlost konvergence a větší rychlost poklesu než u první varianty. Naopak druhá varianta má menší počet iterací než varianta třetí, což ukazuje menší rychlost konvergence u třetí varianty než u varianty druhé.

Uvedeme také grafy konvergenčních historií pro volbu počátečních parametrů $x_0 = -2$ a $\varepsilon = 0,001$.



Obrázek 8: Konvergenční historie subgradientní metody (varianta 3) pro funkci $f(x) = |x|$ s numericky vypočteným gradientem



Obrázek 9: Konvergenční historie subgradientní metody (varianta 3) pro funkci $f(x) = |x|$ s analyticky vypočteným gradientem

Algoritmus pro tuto volbu skončil po 46 iteracích (čímž jsme ilustrovali výsledky z tabulky). Vidíme, že směr kroku d opět nemusí být směr poklesu, ale konvergence je výrazně rychlejší než pro první variantu (první varianta skončí pro stejnou volbu parametrů po 1000 iteracích - převrácená hodnota ε).

5 Metoda r –sečen

V této kapitole si teoreticky popíšeme náležitosti této metody. Metoda r –sečen pracuje se sečnami lokálně lipschitzovských funkcí, které jsou aproximací subgradientů. Při výpočtech budeme hledat aproximaci subgradientu pomocí bundle přístupu, což tuto metodu připodobňuje k bundle metodám. Důkazy všech tvrzení lze nalézt v [1].

5.1 r –sečna

Definice 10 *Nechť $B_1 = \{g \in R^n : \|g\| = 1\}$ je jednotková sféra v R^n . Pro libovolné $g \in B_1$ definujeme*

$$g_{\max} = \max\{|g_i|, i = 1, \dots, n\}. \quad (16)$$

Nechť

$$I(g) = \{i \in \{1, \dots, n\} : |g_i| = g_{\max}\}. \quad (17)$$

Nechť $g \in B_1, r > 0$ a $v \in \partial f(x + rg)$ je libovolný subgradient. Vybereme libovolné $i \in I(g)$. Vektor $s = s(x, g, r) \in R^n$, kde

$$\begin{aligned} s &= (s_1, \dots, s_n) : \\ s_j &= v_j, j = 1, \dots, n, j \neq i \end{aligned} \quad (18)$$

a

$$s_i = \frac{f(x + rg) - f(x) - r \sum_{j=1, j \neq i}^n s_j g_j}{r g_i} \quad (19)$$

se nazývá r –sečnou funkcí f v bodě x ve směru g .

Pokud $n = 1$ definujeme r –sečnu funkce f v bodě $x \in R$ jako

$$s = \frac{f(x + rg) - f(x)}{rg}, \quad (20)$$

kde $g = \pm 1$

Definice 11 *Definujeme množinu*

$$S_{r,g}f(x) = \{s \in R^n : s = s(x, g, r)\}. \quad (21)$$

$S_{r,g}f(x)$ je množina všech možných r –sečen funkce f v bodě x pro pevně zvolené $g \in B^1 \subset R^n$.

Definice 12 Definujeme množinu

$$S_r f(x) = \{s \in R^n : \exists g \in B_1 : s = s(x, g, r)\} = \bigcup_{g \in B_1} S_{r,g} f(x). \quad (22)$$

$S_r f(x)$ je množina všech možných r -sečen funkce f v bodě x . Tato množina je navíc uzavřená a kompaktní pro $r > 0$.

Definice 13 Nyní definujeme následující množiny v bodě $x \in R^n$:

$$S_{0,g} f(x) = \{v \in R^n : \exists \{r_k\} : r_k > 0, \lim_{k \rightarrow \infty} r_k = 0 \text{ a } v = \lim_{k \rightarrow \infty} s(x, g, r_k)\}, g \in B_1, \quad (23)$$

a

$$S_0 f(x) = \bigcup_{g \in B_1} S_{0,g} f(x). \quad (24)$$

$S_{0,g} f(x)$ je množina limitních r -sečen ve směru $g \in B_1$. Množina $S_0 f(x)$ může být definována také jako

$$S_0 f(x) = \{v \in R^n : \exists (g \in B_1, \{r_k\}) : r_k > 0, \lim_{k \rightarrow \infty} r_k = 0 \text{ a } v = \lim_{k \rightarrow \infty} s(x, g, r_k)\}, \quad (25)$$

což je množina limitních r -sečen pro $r \downarrow 0$.

Věta 5 Nechť f je lokálně lipschitzovsky spojitá a necht existují směrové derivace v $x \in R^n$. Pak pro všechna $e = \lambda g$, $\lambda > 0$, $g \in S_1$

$$f'(x, e) = \langle v, e \rangle, \forall v \in S_{0,g} f(x).$$

Věta 6 Nechť f je lokálně lipschitzovsky spojitá a necht existují směrové derivace v $x \in R^n$. Pak pro všechna $e \in R^n$

$$f'(x, e) \leq \max\{\langle v, e \rangle : v \in S_0 f(x)\}.$$

Dodáme ještě definici dvou důležitých množin.

Definice 14

$$S_r^c f(x) = \text{conv } S_r f(x), \quad r > 0, \quad (26)$$

$$S_0^c f(x) = \text{conv } S_0 f(x). \quad (27)$$

Věta 7 Pokud je funkce f polohladká v bodě $x \in R^n$, pak platí

$$S_0^c f(x) \subseteq \partial f(x). \quad (28)$$

Pokud je navíc funkce f v tomto bodě i regulární, platí

$$\partial f(x) = S_0^c f(x). \quad (29)$$

5.1.1 Příklad výpočtu v R

V této podkapitole si na jednoduchém příkladu ukážeme postup výpočtu jednotlivých množin zdefinovaných v předcházejícím textu. Uvedeme postup pro funkci $f(x) = |x|$.

Tato funkce je funkce jedné proměnné, platí pro ni tedy vzorec (20). Budeme počítat r –sečnu v bodě $x = 0$, což je jediný bod nespojitosti první derivace. Dosadíme do vzorce (20) (g je ve všech případech ± 1):

$$s = \frac{f(0 \pm r) - f(0)}{\pm r} = \frac{|\pm r|}{\pm r} = \pm 1 \quad (30)$$

Nyní vypočteme r –sečny pro $x < -r$ dosazením do vzorce

$$s = \frac{f(x \pm r) - f(x)}{\pm r} = \frac{-x \mp r + x}{\pm r} = -1 \quad (31)$$

Dále vypočteme r –sečny pro $x > r$

$$s = \frac{f(x \pm r) - f(x)}{\pm r} = \frac{x \pm r - x}{\pm r} = 1 \quad (32)$$

A obdobně vypočteme i r –sečny pro všechny zbylé případy.

Nyní můžeme sestavit množiny $S_r f(x)$. Ty budou vypadat následovně:

- $S_r f(x) = \{+1, -1\}, x = 0$
- $S_r f(x) = \{+1\}, x > r$
- $S_r f(x) = \{-1\}, x < -r$
- $S_r f(x) = \{1, \frac{2x}{r} - 1\}, x \in \langle 0, r \rangle$
- $S_r f(x) = \{-1, \frac{2x}{r} + 1\}, x \in \langle -r, 0 \rangle$

Z (23) můžeme nyní najít množiny $S_{0,g}f(x)$ pro různou volbu parametrů g .

Pro $g = 1$

$$S_{0,g}f(x) = \begin{cases} \{lim\ 1\} = \{1\} & x = 0 \\ \{lim\ 1\} = \{1\} & x > 0 \\ \{lim\ -1\} = \{-1\} & x < 0 \end{cases}$$

Pro $g = -1$

$$S_{0,g}f(x) = \begin{cases} \{lim\ -1\} = \{-1\} & x = 0 \\ \{lim\ 1\} = \{1\} & x > 0 \\ \{lim\ -1\} = \{-1\} & x < 0 \end{cases}$$

Z (24) víme, že množina $S_0f(x)$ vznikne jako sjednocením všech množin $S_{0,g}f(x)$. Platí tedy, že:

$$S_0f(x) = \begin{cases} \{-1, 1\} & x = 0 \\ \{1\} & x > 0 \\ \{-1\} & x < 0 \end{cases}$$

Nyní už stačí pouze napočítat množiny $S_r^c f(x)$ a $S_0^c f(x)$, které vzniknou jako konvexní obal množin $S_r f(x)$, respektive $S_0 f(x)$. Platí tedy

$$S_r^c f(x) = conv\ S_r f(x) = \begin{cases} \langle -1, 1 \rangle & x = 0 \\ \{1\} & x > r \\ \{-1\} & x < -r \\ \left\langle \frac{2x}{r} - 1, 1 \right\rangle & x \in \langle 0, r \rangle \\ \left\langle -1, \frac{2x}{r} + 1 \right\rangle & x \in \langle -r, 0 \rangle \end{cases}$$

$$S_0^c f(x) = conv\ S_0 f(x) = \begin{cases} \langle -1, 1 \rangle & x = 0 \\ \{1\} & x > 0 \\ \{-1\} & x < 0 \end{cases}$$

Funkce $f(x) = |x|$ je polohladká a regulární. Víme tedy, že množina $S_0^c f(x)$ je zároveň i zobecněným gradientem dané funkce, čímž jsme si také ověřili správnost výsledku příkladu z 3.4.1.

5.1.2 Příklad výpočtu v R^2

Ukážeme si také příklad výpočtu jednotlivých množin pro funkci $f : R^2 \rightarrow R$. V našem případě budeme uvažovat funkci $f(x_1, x_2) = |x_1| + |x_2|$.

Víme, že $B_1 = \{(x_1, x_2) \in R^2 : x_1^2 + x_2^2 = 1\}$ (např. $\{(1, 0); (0, 1); (-1, 0); (0, -1)\} \subset B_1$). Sestrojíme množiny $S_{r,g}, S_r, S_{0,g}, S_0, S_0^c$ v bodě $x = (0, 0)$.

Pro ukázkou výpočtu si zvolíme $g = (1, 0)$. Z (16) a (17) víme, že $i = 1$. Pro r -sečnu s poté platí následující:

$s = (s_1, s_2) :$

$$s_1 = \frac{f(x + r \cdot (1, 0)) - f(x) - r \cdot \sum_{j=1, j \neq 1}^2 s_j g_j}{r \cdot g_1} \quad (33)$$

$$s_2 = v_2, \quad v \in \partial f(x + r \cdot g) = \partial f(r, 0) = \{1\} \times \langle -1, 1 \rangle. \quad (34)$$

Můžeme tedy zvolit libovolné v z množiny určené v (34). V našem případě volíme $v = (1, -1)$ a dosadíme do (33) a (34). Dostaneme následující výsledky:

$$s_2 = v_2 = -1 \quad (35)$$

Odtud vypočteme s_1

$$s_1 = \frac{r - 0 - r \cdot (-1) \cdot 0}{r \cdot 1} = 1 \quad (36)$$

Pro konkrétní volbu $x = (0, 0)$ a $g = (1, 0)$ jsme tedy dostali r -sečnu $s = (1, -1)$. Nyní můžeme přejít ke konkrétnímu výpočtu jednotlivých množin. Z (21) víme, že množina $S_{r,g}f(x)$ je:

$$S_{r,g}f(x) = \{(1, a), \quad a \in \langle -1, 1 \rangle\}.$$

Nyní vypočteme $S_{0,g}f(x)$. Z (23) víme, že množina $S_{0,g}f(x)$ je:

$$S_{0,g}f(x) = \{lim(1, a)\} = \{(1, a), a \in \langle -1, 1 \rangle\}$$

Po dalších výpočtech pro různé volby $g \in B_1$ zjistíme, že se výsledky začnou opakovat. Stejným postupem dostaneme další množiny $S_{r,g}f(x)$. Poté platí:

$$\begin{aligned} S_r f(0,0) &= \bigcup_{g \in B_1} S_{r,g} f(0,0) = \{(1,a); a \in \langle -1,1 \rangle\} \cup \{(-1,a); a \in \langle -1,1 \rangle\} \cup \\ &\cup \{(a,1); a \in \langle -1,1 \rangle\} \cup \{(a,-1); a \in \langle -1,1 \rangle\} \end{aligned}$$

Dostali jsme tedy množinu $S_0 f(0,0)$ z (24) a (25). Odtud už není těžké najít množinu $S_0^c f(0,0)$ pomocí (27).

$$S_0^c f(0,0) = \text{conv}\{(1,1); (-1,1); (1,-1); (-1,-1)\} = \langle -1,1 \rangle \times \langle -1,1 \rangle$$

Víme, že funkce $f(x) = |x_1| + |x_2|$ je polohladká a regulární, platí tedy (29). To znamená, že pro Clarkeův zobecněný gradient v bodě $x = (0,0)$ platí:

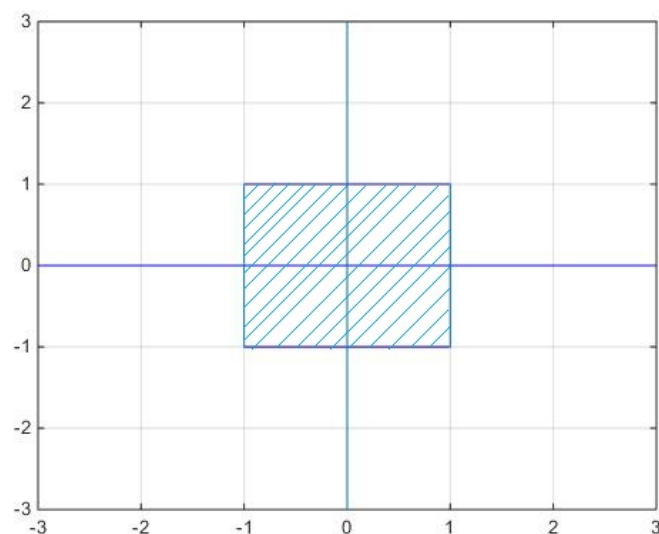
$$\partial f(0,0) = S_0^c f(0,0) = \langle -1,1 \rangle \times \langle -1,1 \rangle,$$

což si můžeme ověřit také pomocí (8):

$$\begin{aligned} \partial f(0,0) &= \text{conv}\left\{ \lim_{(x_1,x_2) \rightarrow (0,0)} \nabla f(x_1,x_2), (x_1,x_2) \notin \Omega_f \right\} = \\ &= \text{conv}\{(1,1); (-1,1); (1,-1); (-1,-1)\} = \langle -1,1 \rangle \times \langle -1,1 \rangle \end{aligned}$$

kde $\Omega_f = \{x \in R^n | f \text{ není diferencovatelná v } x\}$.

Potvrdili jsme si tedy, že platí $\partial f(0,0) = S_0^c f(0,0) = \langle -1,1 \rangle \times \langle -1,1 \rangle$. Pro ilustraci ještě uvedeme množinu $\partial f(x)$ pro $x = (0,0)$.



Obrázek 10: Zobecněný gradient $\partial f(0, 0)$ pro funkci
 $f(x_1, x_2) = |x_1| + |x_2|$

Vidíme, že tedy v bodě $x = (0, 0)$ můžeme vybrat libovolný subgradient z dvojrozměrného intervalu $\langle -1, 1 \rangle \times \langle -1, 1 \rangle$.

5.2 Nutné podmínky minima

V této podkapitole zavedeme pojem tzv. r –stacionárního bodu a zformulujeme nutné podmínky pro minimum funkce. Funkce $f : R^n \rightarrow R$, kterou budeme minimalizovat, je lokálně lipschitzovsky spojitá.

Věta 8 Předpokládejme, že $r > 0$ je dáno a $f(x + rg) \geq f(x)$ pro všechna $g \in B_1$. Pak

$$0_n \in S_r^c f(x) \quad (37)$$

Věta 9 Nechť $x \in R^n$ je lokální minimum funkce f a nechť pro všechna $v \in R^n$ v x existují směrové derivace $f'(x; v)$. Pak platí

$$0_n \in S_0^c f(x).$$

Definice 15 Řekneme, že bod $x \in R^n$ je r –**stacionární bod** (pro $r > 0$) funkce f na R^n , pokud platí, že $0_n \in S_r^c f(x)$.

Nechť $r > 0$. Bod $x \in R^n$ se nazývá (r, δ) –**stacionární** pro funkci f na R^n pokud platí, že $0_n \in S_r^c f(x) + B_\delta$. B_δ označuje sféru se středem v bodě 0_n a poloměrem δ .

Pro bod $x \in R^n$ mohou tedy nastat tři možnosti:

- $x \dots$ stacionární bod $\Rightarrow 0_n \in \partial f(x)$
- $x \dots r$ –stacionární bod $\Rightarrow 0_n \in S_r^c f(x)$
- $x \dots (r, \delta)$ –stacionární bod $\Rightarrow 0_n \in S_r^c f(x) + B_\delta$

Platí, že pokud je bod stacionární, tak je i r –stacionární a z toho plyne, že je i (r, δ) –stacionární (stacionární bod $\Rightarrow r$ –stacionární $\Rightarrow (r, \delta)$ –stacionární).

5.2.1 Příklad

Demonstraci pojmu r –stacionární bod opět provedeme na funkci $f(x) = |x|$. Tentokrát zvolíme $r = 1$. Budeme počítat r –sečny z (20) pro jednotlivé body a pozorovat jak se množina $S_r^c f(x)$ mění.

- $x = 0.5$

$$g = 1 \Rightarrow s = \frac{f(0.5 + 1 \cdot 1) - f(0.5)}{1 \cdot 1} = \frac{1.5 - 0.5}{1} = 1$$

$$g = -1 \Rightarrow s = \frac{f(0.5 + 1 \cdot (-1)) - f(0.5)}{1 \cdot (-1)} = \frac{0.5 - 0.5}{-1} = 0$$

$$S_r f(0.5) = \{0, 1\} \text{ a } S_r^c f(0.5) = \text{conv}\{0, 1\} = \langle 0, 1 \rangle$$

- $x = 1$

$$g = 1 \Rightarrow s = \frac{f(1 + 1) - f(1)}{1} = 1$$

$$g = -1 \Rightarrow s = \frac{f(1 + (-1)) - f(1)}{-1} = 1$$

$$S_r f(1) = \{1\} = S_r^c f(1) = \text{conv}\{1\} = \{1\}$$

- $x = 2$

$$g = 1 \Rightarrow s = \frac{f(2 + 1) - f(2)}{1} = 1$$

$$g = -1 \Rightarrow s = \frac{f(2 + (-1)) - f(2)}{-1} = 1$$

$$S_r f(2) = \{1\} = S_r^c f(2) = \text{conv}\{1\} = \{1\}$$

- $x = -0.5$

$$g = 1 \Rightarrow s = \frac{f(-0.5 + 1 \cdot 1) - f(0.5)}{1 \cdot 1} = \frac{0.5 - 0.5}{1} = 0$$

$$g = -1 \Rightarrow s = \frac{f(-0.5 + 1 \cdot (-1)) - f(-0.5)}{1 \cdot (-1)} = \frac{1.5 - 0.5}{-1} = -1$$

$$S_r f(-0.5) = \{-1, 0\} \text{ a } S_r^c f(-0.5) = \text{conv}\{-1, 0\} = \langle -1, 0 \rangle$$

- $x = -1$

$$g = 1 \Rightarrow s = \frac{f(-1 + 1) - f(-1)}{1} = -1$$

$$g = -1 \Rightarrow s = \frac{f(-1 + (-1)) - f(-1)}{-1} = -1$$

$$S_r f(-1) = \{-1\} = S_r^c f(-1) = \text{conv}\{-1\} = \{-1\}$$

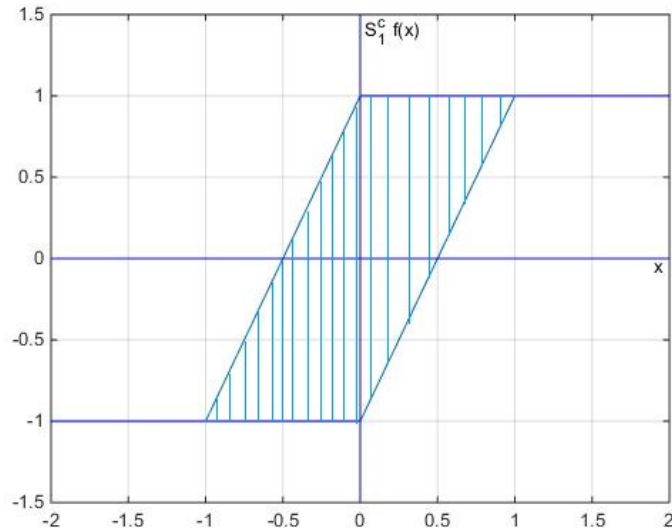
- $x = -2$

$$g = 1 \Rightarrow s = \frac{f(-2 + 1) - f(-2)}{1} = -1$$

$$g = -1 \Rightarrow s = \frac{f(-2 + (-1)) - f(-2)}{-1} = -1$$

$$S_r f(-2) = \{-1\} = S_r^c f(-2) = \text{conv}\{-1\} = \{-1\}$$

Vypočetli jsme tedy množiny S_r^c pro konkrétní body, když $r = 1$. Po dalších výpočtech můžeme vynést průběh těchto množin S_r^c pro všechny body $x \in R$ do následujícího grafu:

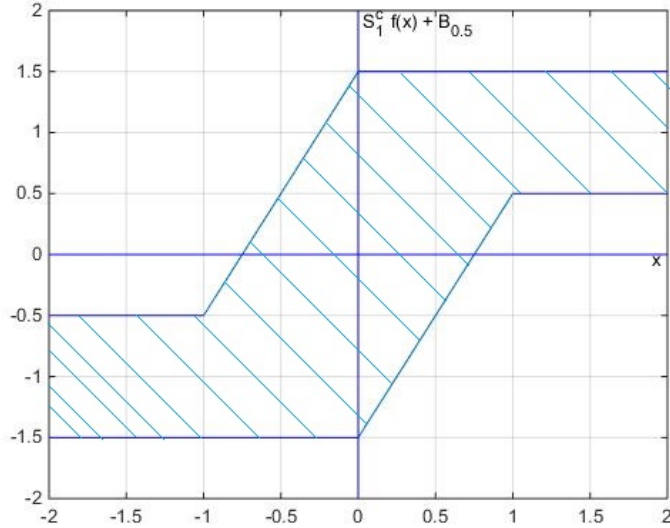


Obrázek 11: Množina $S_1^c f(x)$ pro funkci $f(x) = |x|$

Zde vidíme průběh S_r^c pro všechny body $x \in R^n$, když $r = 1$. Vidíme, že r -stacionární body jsou body z intervalu $x \in \langle -0.5, 0.5 \rangle$, protože pro všechny body tohoto intervalu platí $0_n \in S_r^c f(x)$.

Pro $r = 0$ platí, že se graf změní na graf zobecněného gradientu z příkladu 3.4.1, protože $\partial f(x) = S_0^c f(x)$.

Ukážeme si ještě množinu $S_r^c f(x) + B_\delta$ pro určení (r, δ) –stacionárních bodů. Zvolme opět $r = 1$ a $\delta = 0.5$. Množina $S_r^c f(x) + B_\delta$ s touto volbou vypadá následovně:



Obrázek 12: Množina $S_1^c f(x) + B_{0.5}$ pro funkci $f(x) = |x|$

Z obrázku množiny vyplývá, že pro volbu $r = 1$ a $\delta = 0.5$ nám (r, δ) –stacionární body leží v intervalu $x \in \langle -0.75, 0.75 \rangle$

5.3 Směr poklesu

V této podkapitole ukážeme, že množina $S_0^c f(x)$ může být použita k výpočtu směru poklesu. Pokud víme, že bod $x \in R^n$ není r –stacionárním bodem funkce f na R^n , tzn. že

$$0_n \notin S_r^c f(x), \quad (38)$$

můžeme použít množinu $S_r^c f(x)$ k výpočtu směru poklesu.

Věta 10 *Nechť $f : R^n \rightarrow R$ je lokálně lipschitzovsky spojitá a $r > 0$. Předpokládejme, že $x \in R^n$ není r –stacionárním bodem, tzn. že*

$$\min\{\|v\| : v \in S_r^c f(x)\} = \|v^0\| > 0. \quad (39)$$

Potom pro $g^0 = -\|v^0\|^{-1}v^0$:

$$f(x + rg^0) - f(x) \leq -r\|v^0\|. \quad (40)$$

Toto tvrzení ukazuje, že pokud není $x \in R^n$ r -stacionárním bodem, potom můžeme využít množinu $S_r^c f(x)$ k výpočtu směru poklesu v bodě x . Nicméně konstrukce této množiny není jednoduchá. V dalším textu se budeme zabývat algoritmem, který počítá směr poklesu za použití pouze některých prvků z množiny $S_r^c f(x)$.

Pokud není bod $x \in R^n$ ani (r, δ) -stacionárním bodem, potom $\|v^0\| \geq \delta$. V tomto případě přejde (40) na

$$f(x + rg^0) - f(x) \leq -r\delta. \quad (41)$$

Množina $S_r^c f(x)$ nám poté dovoluje najít směr dostatečného poklesu v bodě $x \in R^n$.

6 Popis algoritmu r –sečen

V této kapitole popíšeme algoritmus pro hledání minima nehladkých funkcí.

Z věty 10 vyplývá, že pro výpočet směru poklesu je třeba řešit následující problém

$$\min \|v\|^2, \quad (42)$$

pro $v \in S_r^c f(x)$. Řešení problému (42) je náročné kvůli výpočtu množiny $S_r^c f(x)$. V následujícím textu popíšeme algoritmus, který používá pouze pár prvků z množiny $S_r^c f(x)$. Podrobnější popis algoritmu může čtenář nalézt v [1].

6.1 Algoritmus 1

Nechť jsou dána čísla $r > 0$, $c \in (0, 1)$ a dostatečně malé $\delta > 0$. Algoritmus 1, který počítá dostatečný směr poklesu obsahuje následující kroky:

1. Vyber nějaké $g^1 \in S_1$ a vypočti r –sečnu $s^1 = s(x, g^1, r)$ ve směru g^1 . Nastav $\overline{W}_1(x) = \{s^1\}$ a $k = 1$.
2. Vypočti $\|w^k\|^2 = \min\{\|w\|^2 : w \in \text{conv}\overline{W}_k(x)\}$. Pokud $\|w^k\| < \delta$ skonči, jinak pokračuj krokem 3.
3. Vypočti směr poklesu $g^{k+1} = -\|w^k\|^{-1}w^k$.
4. Pokud

$$f(x + rg^{k+1}) - f(x) \leq -cr\|w^k\|,$$

skonči, jinak pokračuj krokem 5.

5. Vypočti r –sečnu $s^{k+1} = s(x, g^{k+1}, r)$ ve směru g^{k+1} , sestav $\overline{W}_{k+1}(x) = \text{conv}\{\overline{W}_k(x) \cup \{s^{k+1}\}\}$, nastav $k = k + 1$ a jdi zpět do kroku 2.

V prvním kroku nalezneme r –sečnu v počátečním směru g^1 . Ve druhém kroku hledáme minimální vzdálenost mezi konvexním obalem zatím vypočtených r –sečen (množina $\text{conv}\overline{W}_k(x)$) a počátkem. Jedná se o problém kvadratického programování. Pokud je tato vzdálenost menší, než tolerance $\delta > 0$, je x (r, δ)–stacionární bod. V opačném případě vypočteme ve třetím kroku nový směr, se kterým budeme dále pracovat. Pokud se jedná o směr poklesu, algoritmus skončí nalezením směru poklesu. V opačném případě vypočteme v pátém kroku novou r –sečnu v tomto směru. Takto vylepšíme aproximaci množiny $S_r^c f(x)$. Algoritmus poté hledá směr poklesu s přesnější aproximací množiny $S_r^c f(x)$. Konstantou c řídíme požadovanou velikost poklesu funkční hodnoty funkce f v bodě x ve směru g^{k+1} .

Uvedeme pseudokód pro první část algoritmu.

```

Input  $r, x, f, c \in (0, 1), \delta > 0$ 
 $s_1 = s(x, g^1, r)$ 
 $\overline{W}_0 = s_1$ 
 $k = 1$ 
while true
     $\|w^k\| = \min\{\|w\|^2 : w \in \text{conv}\overline{W}_k(x)\}$ 
    if  $\|w^k\| < \delta$ 
         $g_k = 0$ 
        break
    else
         $g^{k+1} = -\|w^k\|^{-1}w^k$ 
    end;
    if  $f(x + rg^{k+1}) - f(x) \leq -cr\|w^k\|$ 
        break
    else
         $s^{k+1} = s(x, g^{k+1}, r)$ 
         $\overline{W}_{k+1}(x) = \text{conv}\{\overline{W}_k(x) \cup \{s^{k+1}\}\}$ 
         $k = k + 1$ 
    end;
end
return  $g^k, \overline{W}_k(x^k)$ 
end

```

Výpis 4: Algoritmus 1 (metoda r –sečen)

Věta 11 *Předpokládejme, že je funkce f lokálně lipschitzovsky spojitá a $c \in (0, 1), \delta \in (0, M_0)$, kde M_0 je konstanta splňující*

$$\sup\{\|s\| : s \in S_r f(x), x \in X\} \leq M_0,$$

kde $X \subset R^n$ je libovolná omezená podmnožina. Pak platí, že algoritmus 1 má právě m kroků, kde

$$m \leq \frac{2\log_2(\delta/M_0)}{\log_2 M_1} + 2, \quad M_1 = 1 - [(1 - c)(2M_0)^{-1}\delta]^2.$$

6.2 Algoritmus 2

Tento algoritmus je určen k nalezení (r, δ) –stacionárních bodů. Tento algoritmus využívá již popsany algoritmus 1, který hledá směr poklesu v bodě x_k .

Algoritmus 2 vypadá následovně:

Nechť jsou dána čísla $r > 0, \delta > 0, c_1 \in (0, 1), c_2 \in (0, c_1)$.

1. Vyber počteční bod $x^0 \in R$ a nastav $k = 1$.
2. Proveď algoritmus 1 k nalezení dostatečného poklesu v bodě $x = x^k$ pro dané $\delta > 0, c = c_1$. Tento algoritmus končí po $m > 0$ krocích a jeho výsledkem je množina $\overline{W}_k(x^k)$, obsahující prvek v^k takový, že

$$\|v^k\|^2 = \min\{\|v\|^2 : v \in \overline{W}_k(x^k)\}.$$

Navíc platí jedna z možností

- $\|v^k\| \leq \delta$,
- $f(x^k + rg^k) - f(x^k) \leq -c_1 r \|v^k\|$.

3. Pokud $\|v^k\| \leq \delta$, poté skonči, jinak jdi do kroku 4.
4. Sestroj další bod iterace $x^{k+1} = x^k + \alpha_k g^k$, kde α_k je definováno jako

$$\alpha_k = \operatorname{argmax}\{\alpha \geq 0 : f(x^k + \alpha g^k) - f(x^k) \leq -c_2 \alpha \|v^k\|\}.$$

Nastav $k = k + 1$ a jdi do kroku 2.

Jelikož $c_2 \leq c_1$ a délka kroku $\alpha_k \geq r$ a $r > 0$, platí tedy, že r je spodní hranicí pro α_k . To vede k následujícímu pravidlu pro výpočet délky kroku α_k . Zavedeme posloupnost

$$\theta_l = lr, \quad l = 1, 2, \dots$$

Poté α_k je definována jako největší θ_l splňující nerovnost v kroku 4.

Opět uvedeme pseudokód pro tento algoritmus.

```

Input  $r, x, f, c_1 \in (0, 1), c_2 \in (0, c_1), \delta > 0$ 
 $k = 1$ 
while true
     $[g_k, \overline{W}_k(x^k)] = \text{Algoritmus1}(r, x, f, c_1, \delta)$ 
     $\|v^k\|^2 = \min\{\|v\|^2 : v \in \overline{W}_k(x^k)\}$ 
    if  $\|v^k\| \leq \delta$ 
        break
    else
         $\alpha_k = \operatorname{argmax}\{\alpha \geq 0 : f(x^k + \alpha g^k) - f(x^k) \leq -c_2 \alpha \|v^k\|\}$ 
         $x^{k+1} = x^k + \alpha_k g^k$ 
         $k = k + 1$ 
    end
end
return  $x^k$ 
end

```

Výpis 5: Algoritmus 2 (metoda r –sečen)

Věta 12 Pro bod $x_0 \in R^n$ zavedeme množinu $\mathcal{L}(x_0) = \{x \in R^n : f(x) \leq f(x^0)\}$.

Předpokládejme, že funkce f je lokálně lipschitzovsky spojitá a množina $\mathcal{L}(x_0)$ je omezená pro počáteční bod $x_0 \in R^n$. Poté algoritmus 2 nalezne (r, δ) –stacionární bod právě po $K > 0$ iteracích, kde

$$K \leq 1 + \frac{f(x^0) - f_*}{c_2 r \delta}, \quad f_* = \inf\{f(x), x \in R^n\}.$$

6.3 Algoritmus 3

Algoritmus 2 můžeme použít k nalezení Clarkeových stacionárních bodů funkce f . Nechť máme posloupnosti $\{r_k\}, \{\delta_k\}$, pro které platí pro $k \rightarrow \infty$ $r_k \downarrow 0$ a $\delta_k \downarrow 0$. Pokud na tyto posloupnosti aplikujeme algoritmus 2, dostaneme posloupnost $\{x^k\}$, která obsahuje (r_k, δ_k) –stacionární body.

Můžeme tedy zkonstruovat algoritmus 3:

1. Vyber počáteční bod $x^0 \in R^n$ a nastav $k = 1$.
2. Aplikuj algoritmus 2 pro bod x^{k-1} pro $r = r_k$ a $\delta = \delta_k$. Tento algoritmus skončí po konečně mnoha iteracích a jako výsledek nalezne (r_k, δ_k) –stacionární bod x^k .
3. Nastav $k = k + 1$ a jdi do kroku 2

Algoritmus 3 budeme také prezentovat pseudokódem.

```

Input  $r, x, f, f_{opt}, c_1 \in (0, 1), c_2 \in (0, c_1), \delta > 0, \varepsilon > 0$ 
 $x = \text{Algoritmus2}(r, x, f, c_1, c_2, \delta)$ 
while  $\left\| \frac{f_{opt} - f(x)}{1 + |f_{opt}|} \right\| > \varepsilon$  and  $\|r\| > \varepsilon$ 
     $k = k + 1$ 
     $r_{k+1} = 0.6 \cdot r_k$ 
     $x^{k+1} = \text{Algoritmus2}(r, x^k, f, c_1, c_2, \delta)$ 
end
return  $x^k$ 
end

```

Výpis 6: Algoritmus 3 (metoda r –sečen)

V pseudokódu vidíme proměnnou ε , která určuje toleranci pro ukončující podmínku.

Algoritmus skončí, pokud je splněna jedna z následujících částí podmínky:

1. Pokud je $\left\| \frac{f_{opt} - f(x)}{1 + |f_{opt}|} \right\|$ pod hranicí přípustnosti. V tomto zlomku určuje f_{opt} optimální hodnotu funkce v hledaném minimu a $f(x)$ určuje hodnotu funkce v současném bodě iterace. Pokud neznáme f_{opt} , můžeme tuto hodnotu nastavit na $-\infty$.
2. Velikost r je pod hranicí přípustnosti.

V každé iteraci hledáme nový Clarkeův stacionární bod x^{k+1} pomocí algoritmu 2 a poté také musíme změnit velikost proměnné r_{k+1} . V našem případě počítáme $r_{k+1} = 0.6 \cdot r_k$. Proměnnou δ , která určuje (r, δ) –stacionární body, v našem případě volíme jako konstantu.

Předpoklad 1 *Nechť v bodě $x \in R^n$ pro libovolné $\varepsilon > 0$ existuje $\eta > 0$ a $r_0 > 0$ takové, že*

$$S_r^c f(x) \subset \partial f(x + B_\varepsilon) + B_\varepsilon \quad (43)$$

pro všechna $y \in B_\eta(x)$ a $r \in (0, r_0)$. Zde

$$\partial f(x + B_\varepsilon) = \bigcup_{y \in B_\varepsilon(x)} \partial f(y).$$

Věta 13 *Předpokládejme, že funkce f je lokálně lipschitzovsky spojitá a splňuje předpoklad 1 v každém bodě $x \in R^n$ a množina $\mathcal{L}(x_0)$, zavedena ve větě 12, je omezená pro libovolný $x_0 \in R^n$. Poté každý hromadný bod posloupnosti $\{x^k\}$ patří do množiny $X^0 = \{x \in R^n : 0_n \in \partial f(x)\}$.*

7 Numerické experimenty

Po zavedení teoretických pojmů a po detailním popisu v předchozích kapitolách si ukážeme výsledky algoritmu na konkrétních příkladech. Budeme zkoumat výsledné aproximace v závislosti na volbě počátečních parametrů. Jednotlivé vstupní parametry jsou:

- x_0 - počáteční bod (v kapitole 7.3 označujeme jako u_0)
- ε - tolerance pro ukončující podmínku
- r - počáteční velikost pro r -sečnu
- c_1, c_2 - konstanty, kterými řídíme požadovanou velikost poklesu funkční hodnoty
- δ - velikost poloměru, ve kterém hledáme (r, δ) -stacionární body

Výstupy algoritmu jsou počet iterací a x^* , což udává aproximaci hledaného minima.

Pro výpočet subgradientů jednotlivých funkcí opět používáme jak numericky (popsanou ve (14)) tak i analyticky určenou derivaci.

Všechny numerické experimenty byly provedeny v prostředí Matlab.

7.1 S funkcí jedné proměnné

Jako první si algoritmus ukážeme na funkci $f(x) = |x|$, na kterou jsme již aplikovali tři různé varianty subgradientní metody v kapitole 4. Můžeme si tedy kromě přesnosti výsledků také ověřit i rychlost algoritmu vzhledem k jiným metodám.

Budeme řešit úlohu:

$$\min_{x \in R} |x|$$

Pro funkci $f(x) = |x|$ jsme zavedli zobecněný gradient v příkladu 3.4.1. Víme, že minimum funkce je v bodě $x^* = 0$. V následující tabulce vidíme výsledky algoritmu pro různé volby počátečních parametrů.

Tabulka 4: Výsledky metody r –sečen pro funkci $f(x) = |x|$

x_0	ε	r	c_1	c_2	Aproximace minima x^* (subgradient vypočten numericky)		Aproximace minima x^* (subgradient vypočten analyticky)	
					x^*	Počet iterací	x^*	Počet iterací
1	10^{-3}	5	0.5	0.2	$-2.6603 \cdot 10^{-4}$	17	$-2.6603 \cdot 10^{-4}$	17
1	10^{-4}	5	0.5	0.2	$-6.2907 \cdot 10^{-5}$	20	$-6.2907 \cdot 10^{-5}$	20
1	10^{-3}	10	0.5	0.2	$-9.3892 \cdot 10^{-4}$	16	$-9.3892 \cdot 10^{-4}$	16
1	10^{-4}	10	0.5	0.2	$-2.7856 \cdot 10^{-5}$	23	$-2.7856 \cdot 10^{-5}$	23
-2	10^{-3}	5	0.5	0.2	$-2.6603 \cdot 10^{-4}$	17	$-2.6603 \cdot 10^{-4}$	17
-2	10^{-4}	5	0.5	0.2	$-6.2907 \cdot 10^{-5}$	20	$-6.2907 \cdot 10^{-5}$	20
-2	10^{-3}	10	0.5	0.2	$5.3205 \cdot 10^{-4}$	17	$5.3205 \cdot 10^{-4}$	17
-2	10^{-4}	10	0.5	0.2	$-9.3556 \cdot 10^{-5}$	22	$-9.3556 \cdot 10^{-5}$	22

Pro všechny případy v tabulce jsme volili pro optimální funkční hodnotu minima $f_{opt} = 0$ a $\delta = 7 \cdot 10^{-7}$. Z výsledků vidíme, že s menší přesností ε jsme se dostali blíže očekávanému minimu funkce v $x^* = 0$. Aproximace minima i počet iterací pro tuto funkci nezávisí na volbě výpočtu subgradientu (numericky nebo analyticky).

Jak již bylo řečeno, tuto funkci jsme volili kvůli porovnání výsledků algoritmu s výsledky subgradientní metody. V následujících tabulkách uvedeme rozdíly mezi aproximací minima pomocí tří variant subgradientní metody, uvedených v předešlém textu a algoritmem r –sečen v závislosti na volbě počátečních parametrů a výpočtu subgradientu. Každý z algoritmů má různé vstupní parametry, omezíme se tedy pouze na volbu počátečního bodu iterace x_0 a volbu tolerance pro ukončující podmínku ε .

Tabulka 5: Porovnání subgradientní metody a metody r –sečen pro numerický výpočet subgradientu

x_0	ε	Varianta 1		Varianta 2		Varianta 3		Algoritmus r –sečen	
		x^*	Počet iterací	x^*	Počet iterací	x^*	Počet iterací	x^*	Počet iterací
1	0.1	0.0210	10	0.0625	5	-0.0239	11	0.0312	7
1	0.01	$9.1936 \cdot 10^{-5}$	100	0	3	-0.0011	29	-0.0071	11
1	0.001	$-3.999 \cdot 10^{-6}$	1000	$2.4414 \cdot 10^{-4}$	13	$0.0169 \cdot 10^{-3}$	40	$-2.6603 \cdot 10^{-4}$	16
-2	0.1	0.0210	10	-0.0625	6	-0.1878	11	0.0312	7
-2	0.01	$9.1936 \cdot 10^{-5}$	100	$-2.6205 \cdot 10^{-11}$	3	0.0017	26	-0.0071	11
-2	0.001	$-3.999 \cdot 10^{-6}$	1000	$2.4414 \cdot 10^{-4}$	14	$0.0463 \cdot 10^{-3}$	46	$-2.6603 \cdot 10^{-4}$	16

Tabulka 6: Porovnání subgradientní metody a metody r –sečen pro analytický výpočet subgradientu

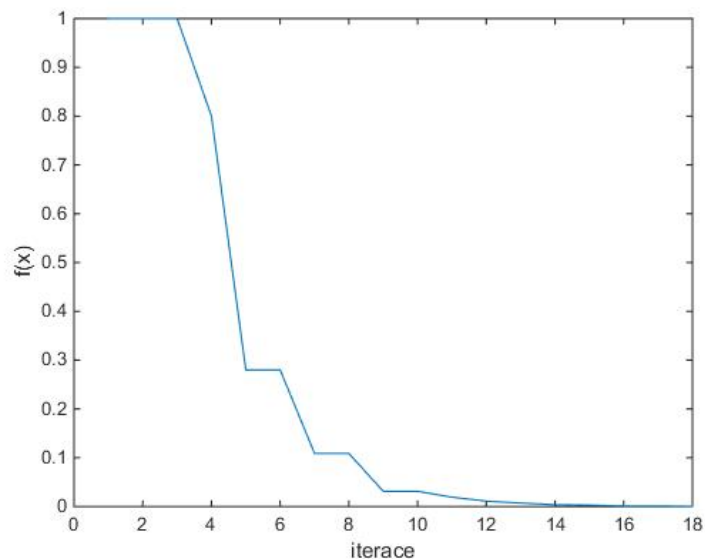
x_0	ε	Varianta 1		Varianta 2		Varianta 3		Algoritmus r –sečen	
		x^*	Počet iterací	x^*	Počet iterací	x^*	Počet iterací	x^*	Počet iterací
1	0.1	0.0210	10	0.0625	5	-0.0239	11	0.0312	7
1	0.01	$-9.6865 \cdot 10^{-5}$	100	0	3	-0.0011	29	-0.0071	11
1	0.001	$-1.0010 \cdot 10^{-6}$	1000	$2.4414 \cdot 10^{-4}$	13	$0.0169 \cdot 10^{-3}$	40	$-2.6603 \cdot 10^{-4}$	16
-2	0.1	0.0210	10	-0.0625	6	-0.1878	11	0.0312	7
-2	0.01	$-9.6865 \cdot 10^{-5}$	100	0	3	0.0017	26	-0.0071	11
-2	0.001	$-1.0010 \cdot 10^{-6}$	1000	$2.4414 \cdot 10^{-4}$	14	$0.0463 \cdot 10^{-3}$	46	$-2.6603 \cdot 10^{-4}$	16

Z tabulek vidíme, že algoritmus r –sečen má méně iterací, než první a třetí varianta subgradientní metody, ale více iterací než druhá varianta. Tento rozdíl ovšem není tak velký. Výsledné

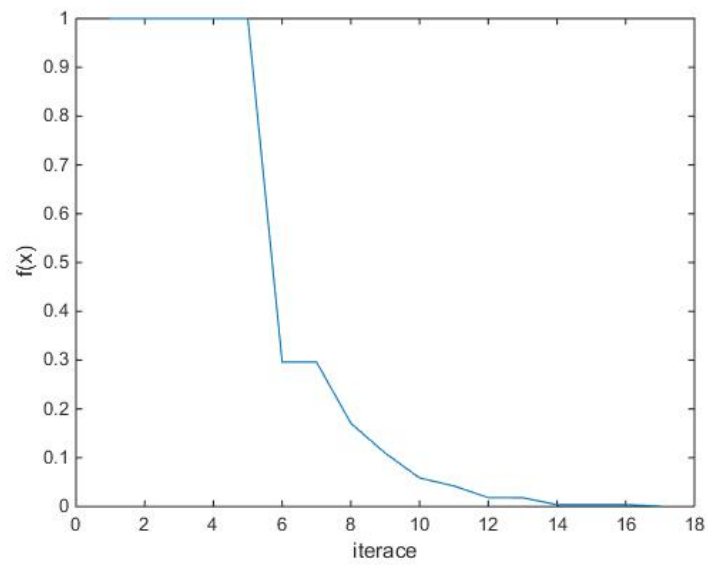
aproximace se liší nejdříve až na druhém desetinném místě, proto se dá říct, že výsledné minimum volba metody neovlivnila. Vzhledem k počtu iterací můžeme ale pro funkci $f(x) = |x|$ navrhnout algoritmus r –sečen jako lepší nástroj pro hledání minima.

Pro porovnání výsledků s předchozími příklady v textu si ještě ukážeme grafy konvergenčních historií pro metodu r –sečen v námi zvolených případech. Tyto grafy ukazují pokles funkční hodnoty v závislosti na počtu iterací.

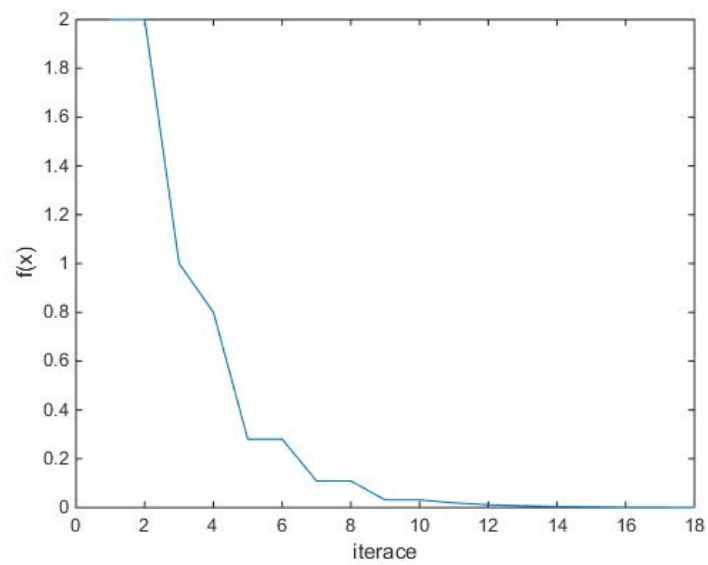
Budeme ilustrovat konvergenční historie pro počáteční parametry z tabulky $x_0 = 1$ a $x_0 = -2$ a pro $r = 5$ a $r = 10$. U všech možností zvolíme $\varepsilon = 10^{-3}$, $c_1 = 0.5$ a $c_2 = 0.2$. Výsledné grafy vidíme na následujících obrázcích.



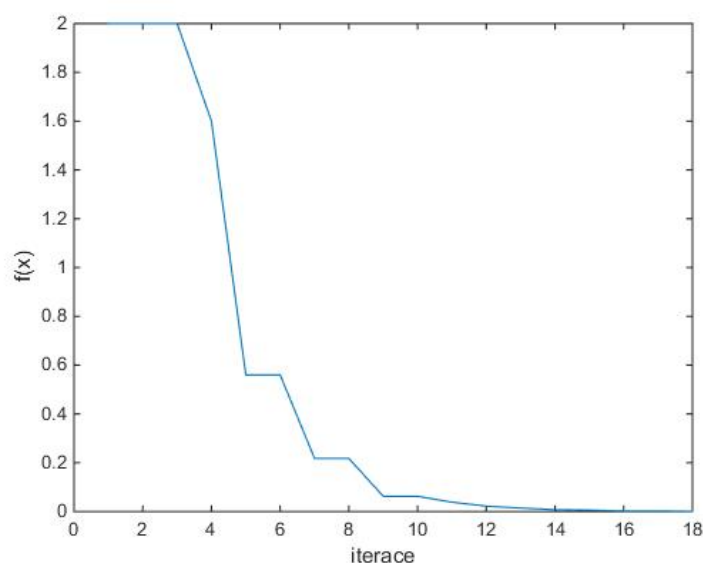
Obrázek 13: Konvergenční historie metody r –sečen pro funkci $f(x) = |x|$ s parametry $x_0 = 1$, $r = 5$



Obrázek 14: Konvergenční historie metody r –sečen pro funkci $f(x) = |x|$ s parametry $x_0 = 1$, $r = 10$



Obrázek 15: Konvergenční historie metody r –sečen pro funkci $f(x) = |x|$ s parametry $x_0 = -2$, $r = 5$



Obrázek 16: Konvergenční historie metody r –sečen pro funkci $f(x) = |x|$ s parametry $x_0 = -2$, $r = 10$

Z grafů vidíme, že se může stát, že se v některé z iterací nezmění bod x^k . To znamená, že bod x^k je (r_k, δ_k) –stacionární pro dané r_k a δ_k . V tom případě algoritmus 2, popsaný v předchozí kapitole, vrací jako výsledek stejný bod, se kterým pracoval v dané iteraci.

Z grafů lze také vidět, že směr, ve kterém hledáme nový bod iterace, je vždy směr poklesu. To je velká výhoda oproti subgradientní metodě, kde směr, ve kterém hledáme další bod iterace, nemusel být vždy směr poklesu.

7.2 S funkcemi dvou proměnných

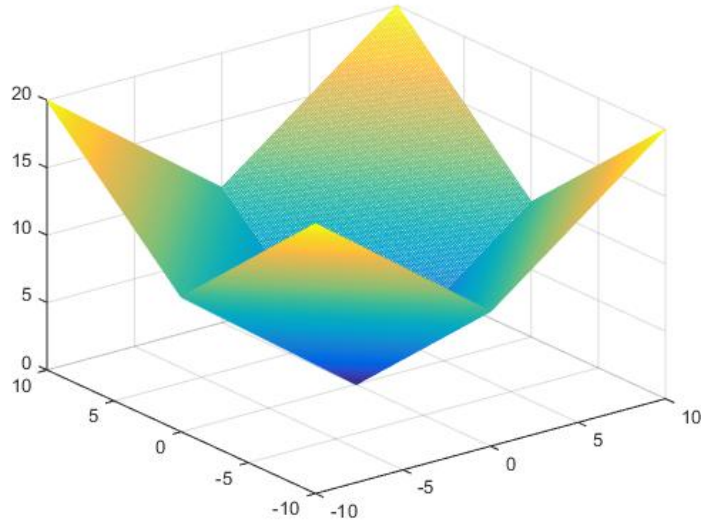
7.2.1 Příklad 1

Budeme řešit úlohu:

$$\min_{x \in \mathbb{R}^2} |x_1| + |x_2|$$

Zobecněný gradient této funkce jsme zavedli v 5.1.2. Víme, že minimum funkce je v bodě $x^* = (0, 0)$ a funkční hodnota je $f(x^*) = 0$.

Na následujícím grafu vidíme, jak vypadá daná funkce a kde se nachází minimum.



Obrázek 17: Graf funkce $f(x_1, x_2) = |x_1| + |x_2|$

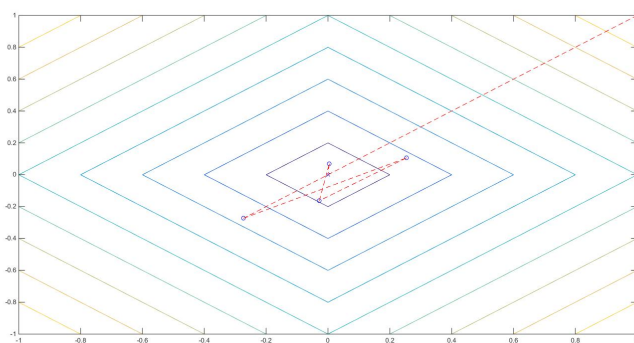
V tabulce vidíme výslednou aproximaci minima a počet iterací, po kterém algoritmus našel řešení. Opět volíme různé počáteční parametry a sledujeme, jak ovlivní tyto volby výsledky. Subgradient počítáme numericky i analyticky, stejně jako v předchozím případě.

Tabulka 7: Výsledky metody r -sečen pro funkci $f(x_1, x_2) = |x_1| + |x_2|$

x_0	ε	r	c_1	c_2	Aproximace minima x^* (subgradient vypočten numericky)		Aproximace minima x^* (subgradient vypočten analyticky)	
					x^*	Počet iterací	x^*	Počet iterací
(1,-2)	10^{-3}	5	0.5	0.2	$(0.2861, 0.2080) \cdot 10^{-3}$	17	$(-0.3213, 0.3828) \cdot 10^{-3}$	16
(1,-2)	10^{-4}	5	0.5	0.2	$(0.2879, 0.0745) \cdot 10^{-4}$	21	$(-0.3688, -0.5716) \cdot 10^{-4}$	21
(1,-2)	10^{-3}	10	0.5	0.2	$(-0.6590, 0.3123) \cdot 10^{-3}$	18	$(0.5550, -0.3062) \cdot 10^{-3}$	18
(1,-2)	10^{-4}	10	0.5	0.2	$(-0.1526, -0.2724) \cdot 10^{-4}$	23	$(0.4388, -0.3252) \cdot 10^{-4}$	22
(-2,3)	10^{-3}	5	0.5	0.2	$(0.1476, 0.1270) \cdot 10^{-3}$	17	$(0.1476, 0.1270) \cdot 10^{-3}$	17
(-2,3)	10^{-4}	5	0.5	0.2	$(-0.1798, -0.2339) \cdot 10^{-4}$	22	$(0.2687, 0.5835) \cdot 10^{-4}$	21
(-2,3)	10^{-3}	10	0.5	0.2	$(-0.3646, -0.4411) \cdot 10^{-3}$	17	$(0.0689, 0.1843) \cdot 10^{-3}$	19
(-2,3)	10^{-4}	10	0.5	0.2	$(0.4785, 0.3894) \cdot 10^{-4}$	22	$(-0.6915, 0.1383) \cdot 10^{-4}$	22

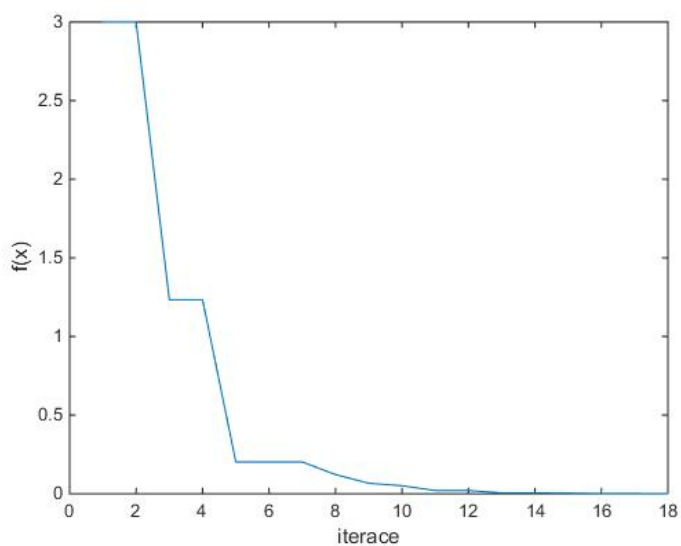
Z tabulky vidíme, že způsob výpočtu subgradientu (numericky, analyticky) má vliv na výslednou aproximaci minima. Počet iterací byl přibližně stejný. V závislosti volby parametru ε se minimum lišilo většinou až na čtvrtém desetinném místě. Ve všech případech jsme volili pro optimální funkční hodnotu minima $f_{opt} = 0$ a $\delta = 7 \cdot 10^{-7}$.

Na následujícím obrázku ilustrujeme, jak konverguje posloupnost x^k k minimu funkce f pro volbu počátečních parametrů $x_0 = (1, 1)$, $r = 5$, $\varepsilon = 0.1$, $c_1 = 0.5$ a $c_2 = 0.2$.

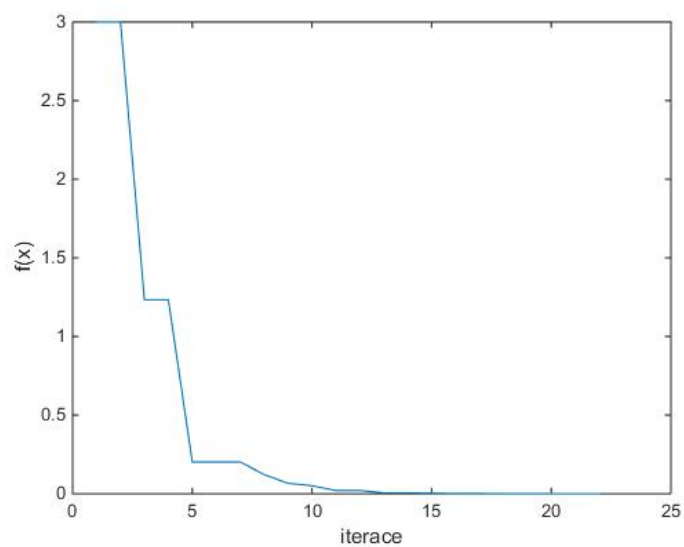


Obrázek 18: Konvergence posloupnosti x^k metody r –sečen pro funkci $f(x_1, x_2) = |x_1| + |x_2|$

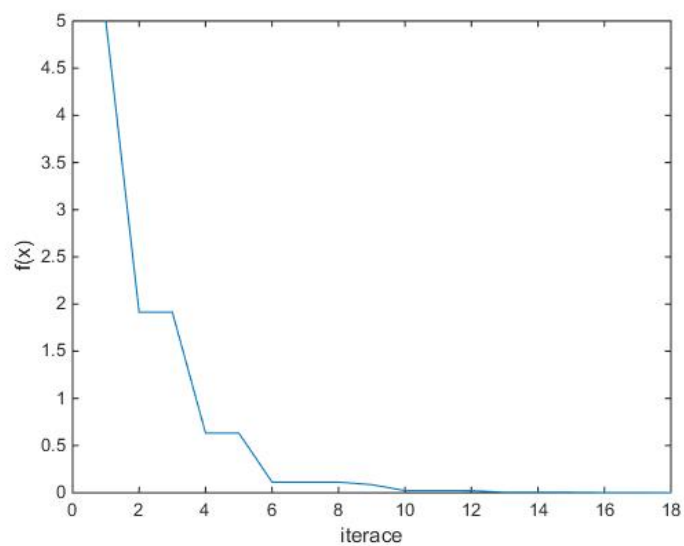
Uvedeme také grafy konvergenčních historií pro některé zvolené případy z tabulky výsledků.



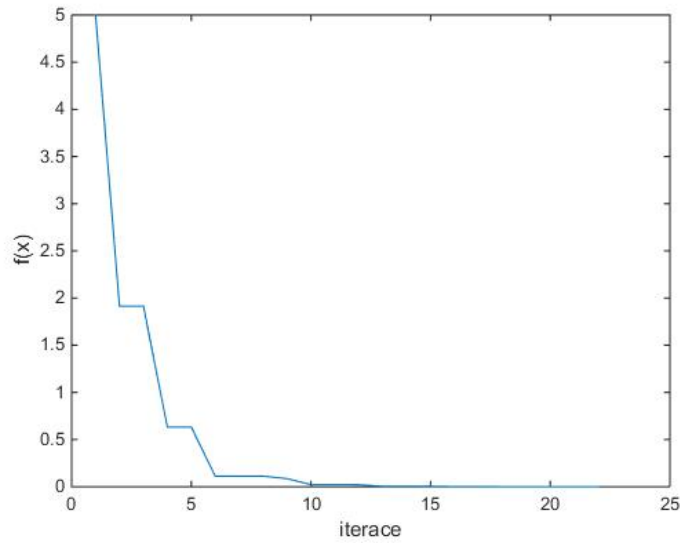
Obrázek 19: Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = |x_1| + |x_2|$ s parametry $x_0 = (1, -2)$, $r = 5$, $\varepsilon = 10^{-3}$



Obrázek 20: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = |x_1| + |x_2|$ s parametry $x_0 = (1, -2)$, $r = 5$, $\varepsilon = 10^{-4}$



Obrázek 21: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = |x_1| + |x_2|$ s parametry $x_0 = (-2, 3)$, $r = 5$, $\varepsilon = 10^{-3}$



Obrázek 22: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = |x_1| + |x_2|$ s parametry $x_0 = (-2, 3)$, $r = 5$, $\varepsilon = 10^{-4}$

Opět vidíme, že v některých iteracích byl nalezen (r_k, δ_k) -stacionární bod pro dané r_k a δ_k , proto iterace zůstala ve stejném bodě x^k jako iterace předchozí.

7.2.2 Příklad 2

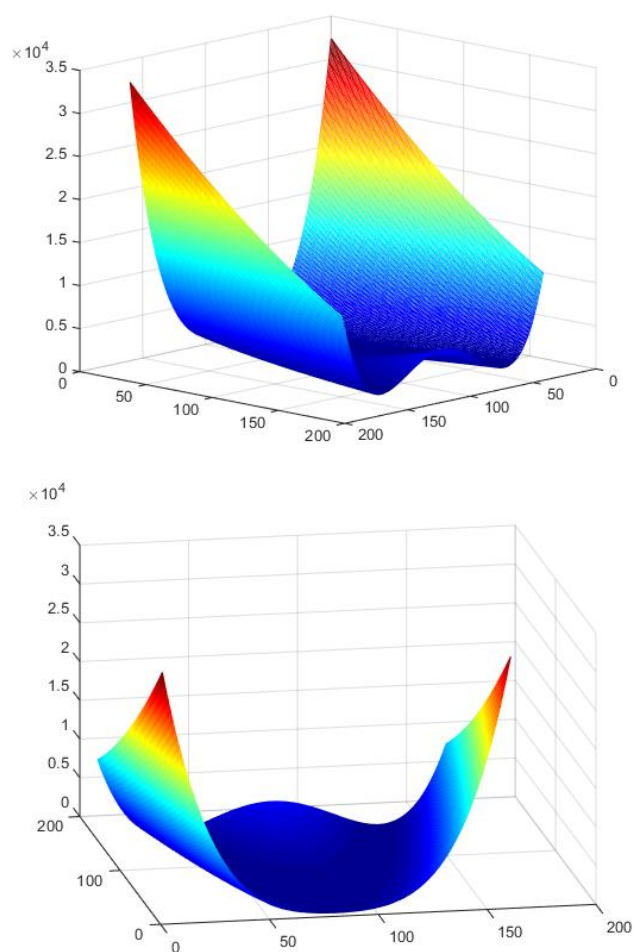
Řešíme úlohu:

$$\min_{x \in \mathbb{R}^2} 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Tato funkce je sice hladká, ale chová se podobně jako funkce nehladké. Její zobecněný gradient je stejný jako gradient, tzn.

$$\partial f(x_1, x_2) = \{\nabla f(x_1, x_2)\} = \begin{bmatrix} 200(x_2 - x_1^2)(-2x_1) + 2(1 - x_1)(-1) \\ 200(x_2 - x_1^2) \end{bmatrix}$$

Minimum, které hledáme, se nachází v bodě $x^* = (1, 1)$ a jeho funkční hodnota je $f(x^*) = 0$. Na následujících obrázcích můžeme vidět grafy této funkce. Lze poznat, že se jedná o funkci hladkou, protože nevidíme žádné body, ve kterých by byly první parciální derivace nespojitě.



Obrázek 23: Graf funkce $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

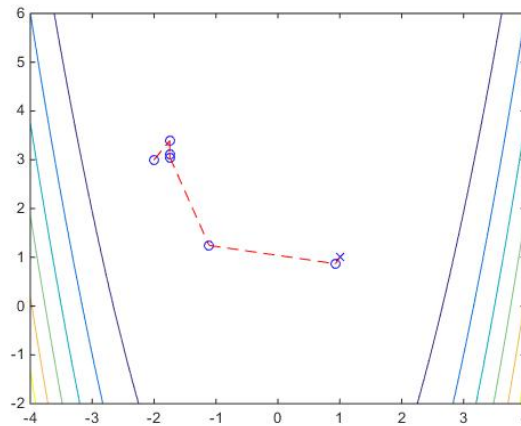
Zde si můžeme ověřit, že algoritmus funguje dobře i pro funkce, které jsou diferencovatelné na celém definičním oboru. Výsledky pro aproximaci minima a počet iterací, po kterém algoritmus našel řešení, můžeme vidět v následující tabulce.

Tabulka 8: Výsledky metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

x_0	ε	r	c_1	c_2	Aproximace minima x^* (subgradient vypočten numericky)		Aproximace minima x^* (subgradient vypočten analyticky)	
					x^*	Počet iterací	x^*	Počet iterací
(1,-2)	10^{-3}	5	0.5	0.2	(1.0153,1.0301)	18	(1.0269,1.0547)	21
(1,-2)	10^{-4}	5	0.5	0.2	(1.0022,1.0045)	24	(1.0074,1.0149)	23
(1,-2)	10^{-3}	10	0.5	0.2	(0.9759,0.9523)	22	(0.9885,0.9771)	24
(1,-2)	10^{-4}	10	0.5	0.2	(0.9915,0.9832)	25	(0.9918,0.9836)	25
(-2,3)	10^{-3}	5	0.5	0.2	(0.9688,0.9384)	23	(0.9864,0.9729)	21
(-2,3)	10^{-4}	5	0.5	0.2	(0.9990,0.9981)	25	(1.0043,1.0087)	24
(-2,3)	10^{-3}	10	0.5	0.2	(0.9750,0.9506)	23	(1.0130,1.0262)	24
(-2,3)	10^{-4}	10	0.5	0.2	(0.9997,0.9994)	25	(1.0039,1.0079)	25

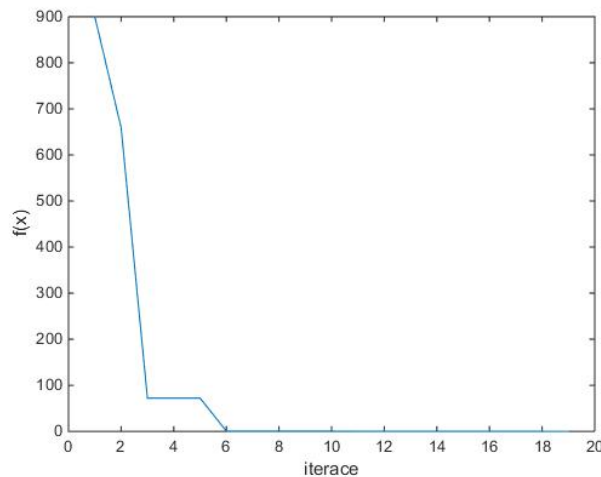
Pro numerické výpočty jsme opět použili hodnoty pro optimální funkční hodnotu minima $f_{opt} = 0$ a $\delta = 7 \cdot 10^{-7}$. Z výsledků vidíme, že se výsledná aproximace blíží očekávanému minimu. Rozdíl v hodnotách je většinou až na třetím desetinném místě. Podle výsledků z tabulky nemůžeme určit, který ze způsobů výpočtu subgradientu je výhodnější, protože výsledné aproximace jsou si velmi podobné a počet iterací, po kterých algoritmus našel výsledek, se také moc neliší.

Na dalším obrázku opět ukážeme konvergenci posloupnosti x^k k očekávanému minimu $x^* = (1, 1)$. Graf odpovídá počátečním hodnotám $x_0 = (-2, 3)$, $r = 10$, $\varepsilon = 0.01$, $c_1 = 0.5$ a $c_2 = 0.2$.

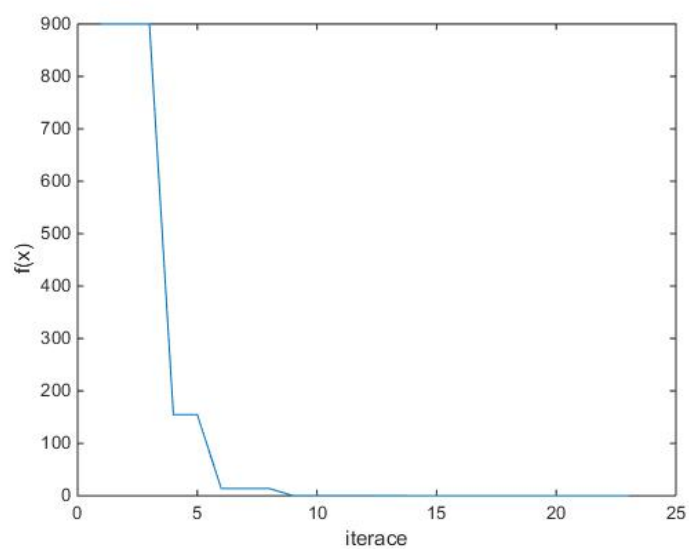


Obrázek 24: Konvergence posloupnosti x^k metody r –sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

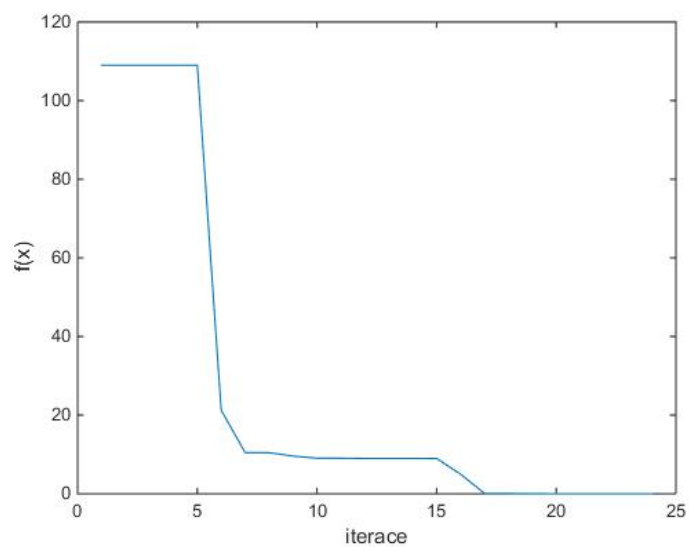
Pro názornost si ještě ukážeme grafy konvergenčních historií. Ve všech případech se subgradient počítá numericky.



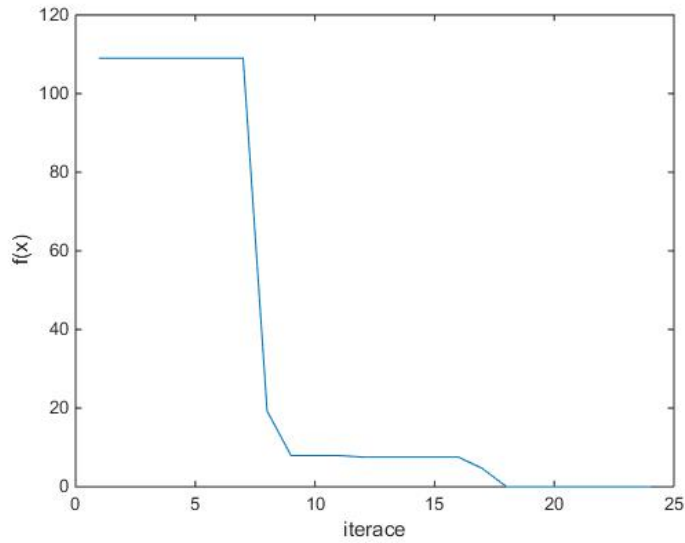
Obrázek 25: Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (1, -2)$, $r = 5$, $\varepsilon = 10^{-3}$



Obrázek 26: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (1, -2)$, $r = 10$, $\varepsilon = 10^{-3}$



Obrázek 27: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (-2, 3)$, $r = 5$, $\varepsilon = 10^{-3}$



Obrázek 28: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ s parametry $x_0 = (-2, 3)$, $r = 10$, $\varepsilon = 10^{-3}$

Na obrázcích vidíme, že se v některých případech opět vyskytly (r_k, δ_k) –stacionární body pro dané r_k a δ_k , proto algoritmus opět zůstal ve stejném bodě. Také vidíme, že v prvních dvou případech se algoritmus dostal velmi rychle do blízkého okolí očekávaného minima.

7.2.3 Příklad 3

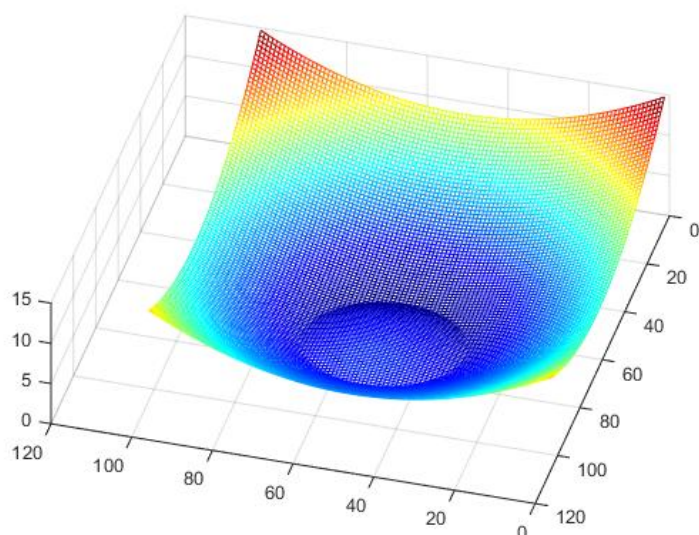
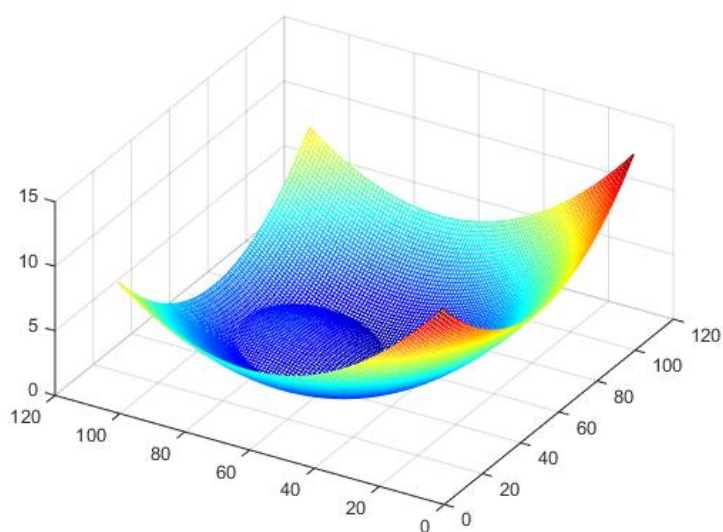
Jako poslední příklad pro funkce v R^2 si ukážeme následující problém:

$$\min_{x \in R^2} \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$$

Zobecněný gradient této funkce vypadá následovně:

$$\partial f(x_1, x_2) = \text{conv} \left\{ \begin{bmatrix} 2x_1 \\ 2x_2 - 1 \end{bmatrix}, \begin{bmatrix} -2x_1 \\ -2x_2 + 3 \end{bmatrix} \right\}$$

Očekávané minimum této funkce se nachází v bodě $x^* = (0, 0)$ a jeho funkční hodnota je $f(x^*) = 0$. Na následujících obrázcích vidíme graf funkce.



Obrázek 29: Graf funkce

$$f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$$

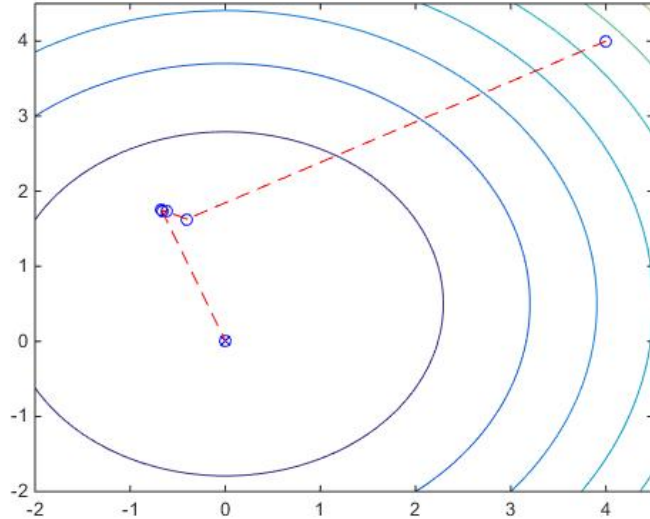
Pro výpočty jsme parametr δ , který určuje poloměr (r, δ) -stacionárních bodů, volili jako konstantu $\delta = 7 \cdot 10^{-7}$. Hodnotu f_{opt} jsme v tomto případě nastavili na $-\infty$ a budeme sledovat, jak tato volba ovlivní výsledné aproximace. Na této funkci si ukážeme, jak se změní celková aproximace při změně konstanty c_1 , která řídí požadovanou velikost poklesu funkční hodnoty. Počáteční bod volíme $x_0 = (4, 4)$ ve všech případech. Tabulka ukazuje výsledné aproximace a počet iterací.

Tabulka 9: Výsledky metody r -sečen pro funkci
 $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$

x_0	ε	r	c_1	c_2	Aproximace minima x^* (subgradient vypočten numericky)		Aproximace minima x^* (subgradient vypočten analyticky)	
					x^*	Počet iterací	x^*	Počet iterací
(4,4)	10^{-3}	5	0.5	10^{-3}	(0.1063,0.0057)	18	(0.1062,0.0056)	18
(4,4)	10^{-3}	5	0.4	10^{-3}	(0.5239,0.1485)	18	(0.5239,0.1485)	18
(4,4)	10^{-3}	5	0.3	10^{-3}	$(-0.3918, -0.0019) \cdot 10^{-3}$	18	$(0.5485, 0.0002) \cdot 10^{-3}$	18
(4,4)	10^{-3}	5	0.2	10^{-3}	(-0.0504,0.0013)	18	(-0.0504,0.0013)	18
(4,4)	10^{-3}	10	0.5	10^{-3}	(0.3004,0.0461)	20	(0.3004,0.0462)	20
(4,4)	10^{-3}	10	0.4	10^{-3}	(0.2734,0.0381)	20	(0.2727,0.0379)	20
(4,4)	10^{-3}	10	0.3	10^{-3}	(0.2769,0.0390)	20	(0.2769,0.0391)	20
(4,4)	10^{-3}	10	0.2	10^{-3}	(0.6488,1.7610)	20	(0.6488,1.7610)	20

Z výsledků vyplývá, že aproximace minima je závislá na volbě parametru r , protože jsme volili $f_{opt} = -\infty$. Algoritmus totiž pouze ověřuje pokud není hodnota r pod hranicí přípustnosti (kritérium zavedené v 6.3). Výsledné aproximace a počty iterací jsou u obou variant výpočtu subgradientu téměř totožné. U třetího řádku v tabulce vidíme, že odpovídá očekávanému minimu nejlépe vzhledem k volbě počátečních parametrů. Naopak u posledního řádku si můžeme všimnout, že nalezené minimum je velmi vzdálené od očekávaného, a proto bychom v tomto případě neměli vynechat první část ukončující podmínky.

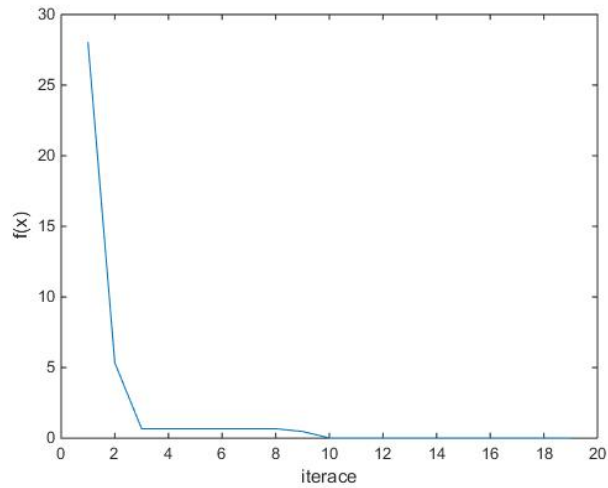
Ukážeme si také konvergenci posloupnosti x^k k očekávanému minimu $x^* = (0, 0)$ pro třetí řádek tabulky výsledků, tedy pro počáteční parametry $x_0 = (4, 4)$, $\varepsilon = 10^{-3}$, $r = 5$, $c_1 = 0.3$, $c_2 = 10^{-3}$.



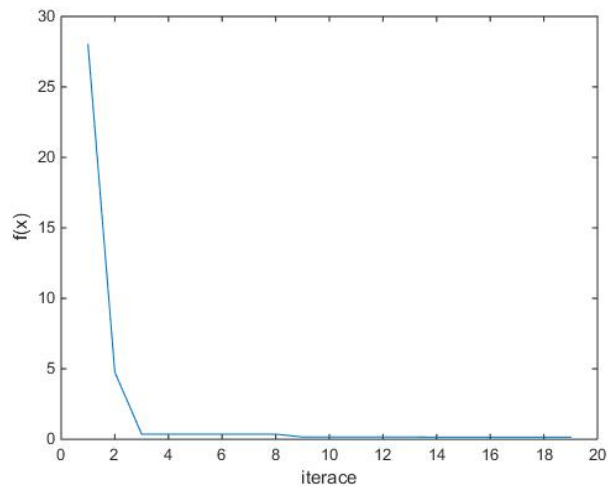
Obrázek 30: Konvergence posloupnosti x^k metody r -sečen pro funkci
 $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$

Jak již bylo řečeno, tento příklad ilustruje, jak moc ovlivní změna parametru c_1 výslednou aproximaci. Tato konstanta řídí požadovanou velikost poklesu funkční hodnoty. Rozdíl v jednotlivých případech se nejvíce projeví na grafech konvergenčních historií, ty totiž popisují závislost počtu iterací a právě funkčních hodnot současného bodu iterace.

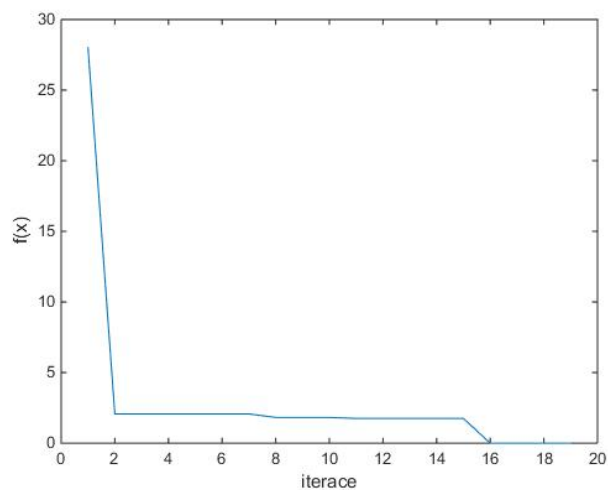
Následující obrázky konvergenčních historií odpovídají vstupním parametrům $x_0 = (4, 4)$, $\varepsilon = 10^{-3}$, $r = 5$ a $c_2 = 10^{-3}$. Měnili jsme pouze konstantu c_1 .



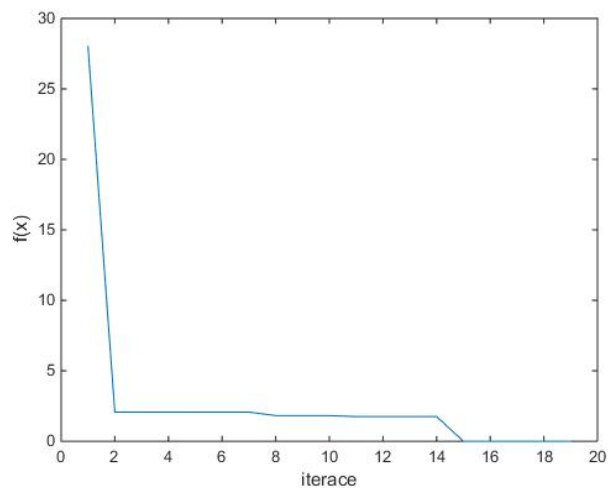
Obrázek 31: Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.5$



Obrázek 32: Konvergenční historie metody r –sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.4$



Obrázek 33: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.3$



Obrázek 34: Konvergenční historie metody r -sečen pro funkci $f(x_1, x_2) = \max[x_1^2 + (x_2 - 1)^2 + x_2 - 1, -x_1^2 - (x_2 - 1)^2 + x_1 + 1]$ s parametrem $c_1 = 0.2$

Vidíme, že v prvním a druhém případě se algoritmus dostal rychle do blízkého okolí minima. Byl tak zaručen rychlejší pokles funkční hodnoty pro větší konstantu c_1 . Pro menší c_1 se algoritmus dostal do blízkého okolí minima až v 15, respektive 16 iteraci.

7.3 S pružnou membránou

Jako poslední příklad si ukážeme modelovou úlohu prohnutí membrány. Tato úloha má konkrétní fyzikální význam. Uvažujeme pružnou membránu a navíc odpor prostředí na její hranici, který popisujeme podobně jako tření v úlohách mechaniky. Podrobnější informace o zadání úlohy lze nalézt v [5] a informace o diskretizaci úlohy lze nalézt v [2] a [6].

7.3.1 Zadání

Budeme řešit úlohu:

$$\min J(u), \text{ kde } J(u) = \frac{1}{2} a(u, u) - (f, u) + j(u),$$

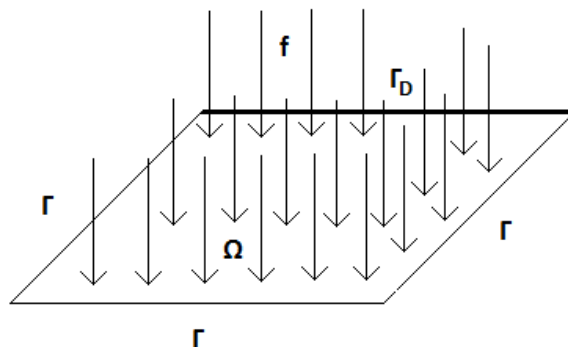
$$\text{s omezením } u = 0 \text{ na } \Gamma_D, \text{ kde}$$

$$a(u, v) = \int_{\Omega} \left(uv + \sum_{i=1}^n \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \right) d\Omega,$$

$$(f, u) = \int_{\Omega} f \cdot u d\Omega,$$

$$j(u) = \int_{\Gamma} g|u| d\Gamma.$$

Úlohu řešíme na membráně, která je popsána množinou $\Omega = \langle 0, 1 \rangle \times \langle 0, 1 \rangle$. Tato membrána je pevně uchycena na hranici Γ_D . Na membránu působíme silou $f = -1$. Na hranici Γ působí na membránu odpor prostředí, který popisujeme funkcí j . V diskrétním tvaru lze prohnutí membrány v každém bodě sítě popsat jedním vektorem u (složky u jsou prohnutí v jednotlivých bodech sítě). Dimenze úlohy je tedy rovna počtu uzlů sítě. Situace popisující sílu působící na membránu je zobrazena na následujícím obrázku.



Obrázek 35: Popis pružné membrány

7.3.2 Řešení

Nejprve musíme provést diskretizaci sítě pomocí metody konečných prvků. Po diskretizaci řešíme problém minimalizace nehladkého funkcionálu

$$J(u) = \frac{1}{2} u^T A u - u^T f + |b^T u|$$

Tento problém již řešíme pomocí metody r –sečen, která byla vysvětlena v předchozích kapitolách.

7.3.3 Diskretizace pomocí konečných prvků

Nyní si ukážeme výsledné vztahy pro diskretizaci pomocí metody konečných prvků. Necht existuje diskretizace oblasti $\Omega = \langle 0, 1 \rangle \times \langle 0, 1 \rangle$ na trojúhelníkové elementy T . Takovýto element T je daný třemi body, jejichž souřadnice jsou $[x_i, y_i], [x_j, y_j], [x_m, y_m]$. Prohnutí elementu T označíme jako δ_T .

Výsledná diskretizace složek funkcionálu $J(u)$ zavedeného v předchozím textu vypadá následovně:

$$a(u, u) = \sum_{el.} \delta_T^T \cdot A_T \cdot \delta_T,$$

kde A_T je lokální matice tuhosti popsána:

$$A_T = \frac{1}{2} \cdot \Delta \cdot \left(\frac{1}{3} I + K^T K \right), \Delta = \det \begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{bmatrix}, K = \begin{bmatrix} y_j - y_m & y_m - y_i & y_i - y_j \\ x_m - x_j & x_i - x_m & x_j - x_i \end{bmatrix},$$

a I je jednotková matice.

$$(f, u) = \sum_{el.} \delta_T^T \cdot F_T,$$

kde F_T je lokální vektor zatížení:

$$F_T = \frac{1}{2} \cdot \Delta \cdot \int_T f \cdot (1, x, y) dT \cdot N^T,$$

kde f je síla působící na membránu (v našem případě $f = -1$) a

$$N = \frac{1}{\Delta} \begin{bmatrix} x_j y_m - x_m y_j & x_m y_i - x_i y_m & x_i y_j - x_j y_i \\ y_j - y_m & y_m - y_i & y_i - y_j \\ x_m - x_j & x_i - x_m & x_j - x_i \end{bmatrix}.$$

Nechť existuje diskretizace hranice Γ na elementy L . Mějme jeden element L daný dvěma body se souřadnicemi $[x_i], [x_j]$. Označme prohnutí elementu L jako δ_L . Poté platí:

$$j(u) = \sum_{el.} g \cdot |P_L^T \cdot \delta_L|,$$

kde g je kladný koeficient (v úlohách mechaniky g popisuje tření) a

$$P_L = \begin{bmatrix} \frac{\Delta}{2} \\ \frac{\Delta}{2} \end{bmatrix}, \Delta = x_j - x_i.$$

Globální matici tuhosti, globální vektor zatížení a globální vektor P poté složíme z lokální matice tuhosti, lokálního vektoru zatížení a lokálního vektoru P . Dostáváme tedy potřebný funkcionál $J(u) = \frac{1}{2} u^T A u - u^T f + |b^T u|$, který již můžeme minimalizovat pomocí metody r -sečen.

7.3.4 Výsledky

Pro výsledné měření volíme počáteční parametry $r = 1$, $c_1 = 0.1$, $c_2 = 0.005$, $\delta = 1$ a $\varepsilon = 10^{-3}$. Parametr r stejně tak jako parametr δ v každé iteraci měníme pomocí vztahu $r_{k+1} = 0.6 \cdot r_k$, resp. $\delta_{k+1} = 0.6 \cdot \delta_k$ (δ_k volíme jinak než v předchozím textu). Jako počáteční vektor u_0 volíme vektor obsahující stejný počet jedniček jako počet uzlů v síti. Neznáme optimální funkční hodnotu funkce v hledaném minimu, proto volíme $f_{opt} = -\infty$.

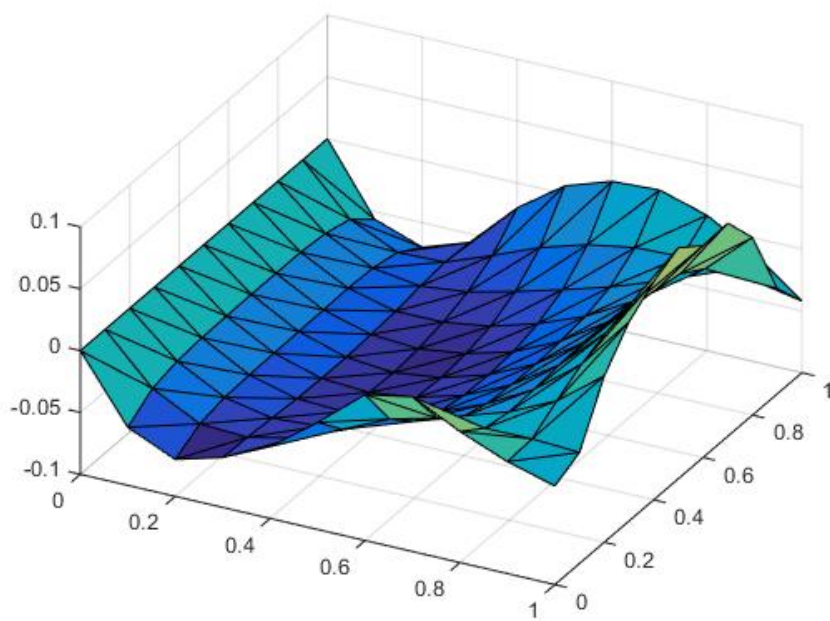
Vzhledem k časové náročnosti algoritmu volíme diskretizaci sítě na 10x10 bodů. Dostáváme tak funkci o 121 proměnných.

Ukážeme si výsledky a prohnutí membrány pro různé volby koeficientu g ($g = 1$, $g = 0.3$ a $g = 0.1$). V tabulce vidíme počet iterací i hodnotu funkcionálu v minimálním bodě.

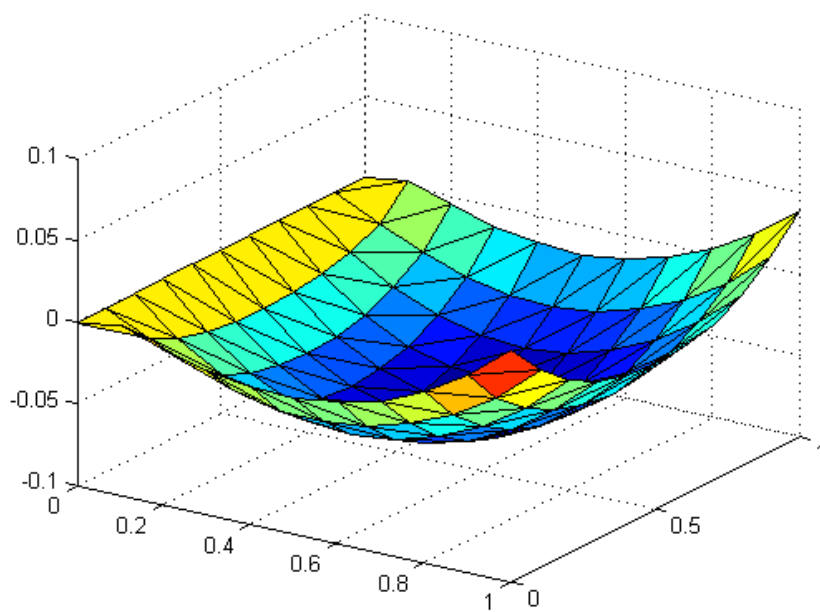
Tabulka 10: Výsledky metody r -sečen pro řešení úlohy prohnutí membrány

ε	g	Počet iterací	$J(u)$
10^{-3}	1	15	0.0040
10^{-3}	0.3	15	-0.0169
10^{-3}	0.1	15	-0.0728

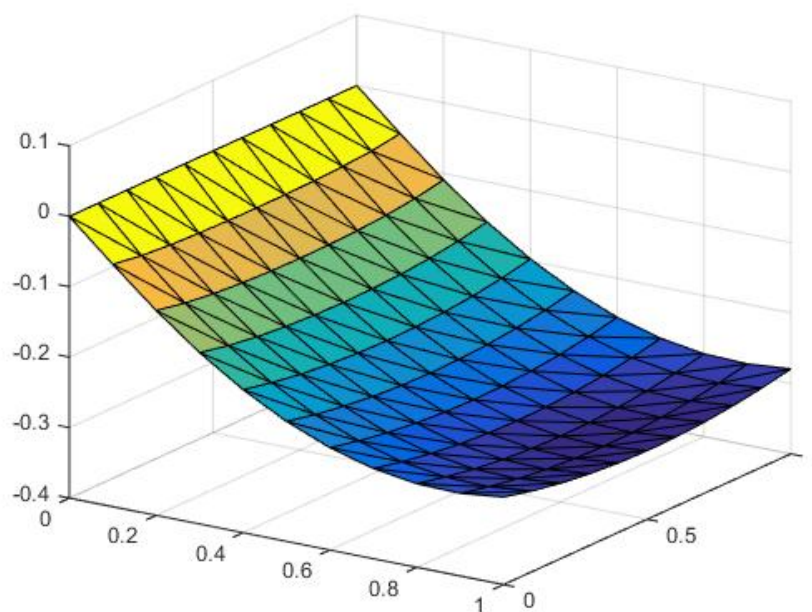
Počet iterací se nemění, protože jsme nastavili hodnotu $f_{opt} = -\infty$. Ukončující podmínka je dána pouze velikostí r_k . Odtud stejný počet iterací. Vidíme, že se výsledná hodnota funkcionálu pohybuje okolo nuly.



Obrázek 36: Prohnutí membrány pro $g = 1$



Obrázek 37: Prohnutí membrány pro $g = 0.3$



Obrázek 38: Prohnutí membrány pro $g = 0.1$

Z obrázků vidíme, jak velký vliv má koeficient g na výsledné prohnutí membrány. Pro $g = 1$ je odpor na hranici natolik velký, že začne membránu v okolí hranice zvedat. Pro volbu $g = 0.3$ vidíme, že je odpor na hranici výrazně menší. Membrána se prohne, ale již se na hranici nezvedá o tolik jako v předchozím případě. Pro volbu $g = 0.1$ je odpor téměř nulový, nemá tedy skoro žádný vliv na výsledné prohnutí. Membrána se lehce zvedá pouze na hranici, kde působí odpor prostředí.

Dokázali jsme si také, že metoda r –sečen je vhodná pro minimalizaci funkce více proměnných (v našem případě funkce 121 proměnných). Rychlost a efektivnost algoritmu závisí na volbě počátečních parametrů.

8 Závěr

Hlavním cílem práce bylo seznámení se s metodou r –sečen, její následná implementace a použití na vhodných úlohách. Tento cíl byl splněn a metoda r –sečen byla implementována v Matlabu. Všechny experimenty byly také provedeny v Matlabu.

V teoretické části jsme nejdříve zavedli pojmy potřebné pro zavedení iteračních metod určených pro minimalizaci nehladkých funkcí.

Poté následoval popis tří variant subgradientní metody společně s numerickými testy. Zde jsme zjistili, že subgradientní metoda nemusí být ideální optimalizační nástroj. Směr, ve kterém hledáme nový bod iterace, totiž nemusí být vždy směrem poklesu, což jsme si ilustrovali i na grafech konvergenčních historií. Tyto algoritmy také většinou dosahují většího počtu iterací, protože neexistuje ideální ukončující kritérium.

Poté jsme se již zabývali teoretickým popisem algoritmu r –sečen a numerickými testy. V první části numerických testů jsme porovnávali rozdíly ve výsledcích mezi algoritmem r –sečen a třemi variantami subgradientní metody. Zjistili jsme, že algoritmus r –sečen našel většinou výsledek po menším počtu iterací. Výsledná aproximace se lišila nejdříve až na druhém desetinném místě. Zde jsme navrhli algoritmus r –sečen jako lepší nástroj pro hledání minima funkce jedné proměnné.

Na funkcích dvou proměnných jsme zjistili, že algoritmus funguje i pro diferencovatelné funkce. Zde jsme si ukázali i grafy popisující, jak konverguje posloupnost iteračních bodů generovaných metodou r –sečen k hledanému minimu. Z grafů konvergenčních historií jsme zjistili, že směr, ve kterém hledáme nový iterační bod, je vždy směrem poklesu. Mohlo se stát, že algoritmus „zůstal“ v některých iteracích ve stejném bodě. V tomto případě to znamenalo, že tento bod byl (r_k, δ_k) –stacionárním bodem pro dané r_k a δ_k .

V poslední části jsme poté prováděli numerické experimenty s pružnou membránou, na kterou působil na hranici odpor prostředí. Zde jsme kvůli časové náročnosti volili diskretizaci sítě krokem $h = 0.1$. Výsledná hodnota funkcionálu se pohybovala kolem 0. Zjistili jsme, že časová náročnost a efektivita je závislá na volbě počátečních parametrů. Při vhodné volbě parametrů metoda r –sečen dokáže vyřešit i praktické optimalizační úlohy.

Literatura

- [1] BAGIROV, Adil a A. Nazari GANJEHLOU. *A secant method for nonsmooth optimization*. Frontiers Science Series, 2007, 49:243.
- [2] BEREMLIJSKI, Petr. *On Solving a Contact Problem with Coulomb Friction by Bundle Trust Method*. Transactions of the VŠB - Technical University of Ostrava, Vol. 1 (2001), 11-20.
- [3] CLARKE, Frank H. *Optimization and nonsmooth analysis*. New York: Wiley, 1983. ISBN 047187504x.
- [4] DOSTÁL, Zdeněk a Petr BEREMLIJSKI. *Metody optimalizace*. [online]. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/metody_optimalizace.pdf. 2012.
- [5] GLOWINSKI, Roland, JACQUES L. LIONS a R. TREMOLIERES. *Analyse Numérique des Inéquations Variationnelles*. Ruský překlad - Moskva, 1979
- [6] ZIENKIEWICZ, O. C. *The finite element method in engineering science*. New York: McGraw-Hill, 1971. ISBN 0070941386.
- [7] ZOWE, J. *Nondifferentiable Optimization*. NATO ASI Series, 15, Computational Mathematical Programming, Springer Berlin Heidelberg, Berlin, Heidelberg, 1985, 323-356. ISBN 9783642824500.